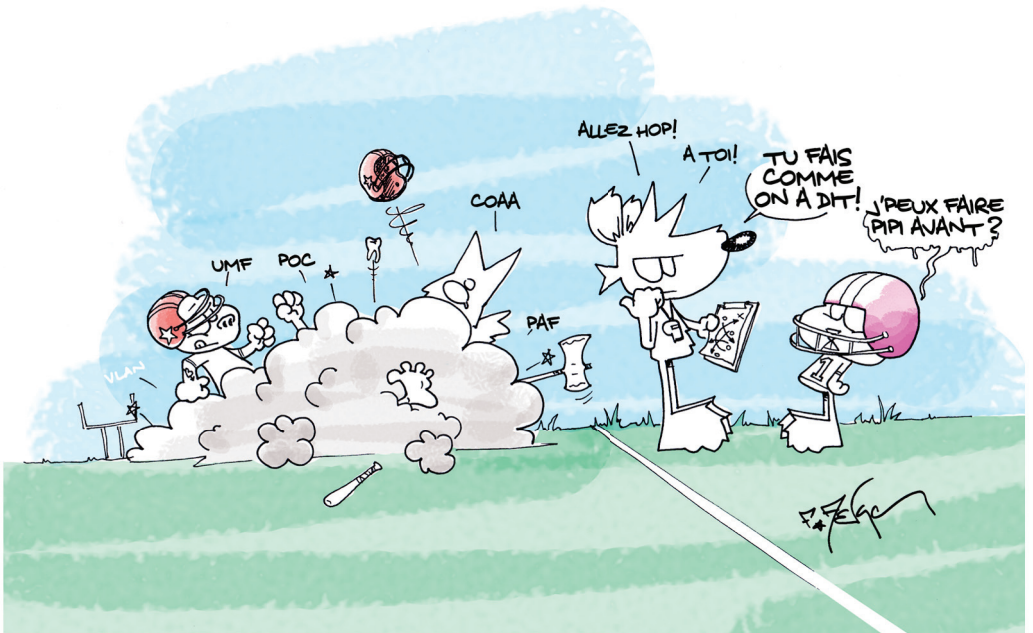


Powered by Xebia

SCRUM MASTER ACADEMY

Le guide du Scrum Master d'élite



Gilles Mantel

Scrum Master Academy

Le guide du Scrum Master d'élite

© 2013 Gilles Mantel

A propos de l'auteur : Gilles Mantel

Gilles est un consultant senior, directeur de projets agiles de plus de 10 ans d'expérience. Il est impliqué sur des projets itératifs depuis plus de 7 ans. Depuis 5 ans, il accompagne l'adoption des pratiques agiles et conseille sur les méthodologies pilotées par les tests (TDR, ATDD). Ses compétences sur les méthodes agiles et les tests sont reconnues et il participe régulièrement en tant qu'orateur à des conférences. Gilles est responsable de l'offre agile chez Xebia depuis plus de deux ans.

TWEET THIS BOOK !

Please help Gilles Mantel by spreading the word about this book on Twitter !

The suggested tweet for this book is :

[I just bought #ScrumMasterAcademy by @gmantel](#)

The suggested hashtag for this book is

[#scrummasteracademy.](#)

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter :

<https://twitter.com/search/#scrummasteracademy>

TABLE DES MATIÈRES

Preface	5
Introduction	7
Parlons de valeurs	10
Règle n°1 : Connais tes valeurs agiles	11
Règle n°2 : Le manifeste agile n'est pas la déclaration des droits du développeur	14
Règle n°12 : On travaille toujours en équipe	16
Règle n°19 : No broken window	19
Règle n°21 : Évalue régulièrement où l'équipe en est de sa maîtrise de l'agile	20
Parlons des user stories	22
Règle n°4 : N'oublie jamais la valeur d'une story	24
Règle n°5 : Une story trop grosse pour entrer dans un sprint est une Epic	26
Règle n°6 : Il faut pouvoir prendre plus de 4 stories par sprint.....	28
Règle n°8 : À la fin du sprint, il y a 2 options pour une story : « done » ou « not done ».....	30
Règle n°9 : Tu n'estimeras pas en jour/homme	32
De l'efficacité des cérémonies Scrum	33
Règle n°3 : Le burndown est publié tous les jours après le Stand Up	34
Règle n°30 : Démon sans slides.....	37
Règle n°31 : La démo commence à l'heure et avec tout le matériel adéquat	38
Règle n°34 : Tu connais et tu lis les stories avant le sprint planning	39
Règle n°15 : En rétrospective, varier les formats	40
Règle n°36 : Favorise les TIG pour gérer les retard en Daily Scrum	41
Règle n°25 : Toujours avoir des post-its sur soi	42
La résolution des obstacles	43
Règle n°14 : Ne te laisse pas enterrer sous les problèmes	44
Règle n°20 . Il n'y a pas de «eux», il y a «nous»	46

Règle n°22 . Connais ton ennemi.....	48
Règle n°26 . Ne rien considérer comme allant de soi.....	49
Règle n°27 . Il vaut mieux demander pardon qu'obtenir la permission	50
Règle n°35 . Un Scrum Master sur le carreau est inutile	51

L'attitude et le charisme	52
Règle n°7 : L'énergie de l'équipe c'est toi qui l'apporte	53
Règle n°10 : Dans une réunion, ta place est debout à côté du tableau blanc.....	56
Règle n°18 : L'auto-organisation nécessite du Leadership....	58
Règle n°29 : Tu es un facilitateur pas un dictateur	60
Règle n°13 : Reste à la page	61
Règle n°32 : Évite le blâme et la complaisance	62
Règle n°16 : Tu portes la transparence	63
Règle n°23 : Écoute et tais toi.....	64

Allez plus loin !	65
Règle n°24 : Veille à la pluralité des compétences de l'équipe	66
Règle n°28 . Ne croyez jamais ce qu'on vous dit, vérifiez toujours	67
Règle n°11 : Ne deviens pas un garagiste.....	68
Règle n°33 : Tu déploies tous les jours en environnement de tests.....	69

Règles de crise	70
Règle n°17 : Transparence et pas indécence	71
Règle n°40 : Les points de story sont une fourchette de jours/ hommes et un % de certitude	72
Règle n°41 : Gère les pollueurs en démo.....	73
Règle n°42 : Range ton désordre	74
Règle n°43 : Pas de sprint entier de refactoring	75
Règle n°44 : Si vous êtes en Scrum distribué, tout le monde sur le même site	76
Règle n°45 : À la rétrospective, 2 actions d'amélioration maximum	77

Vos règles du jeu	79
--------------------------------	----

Conclusion et remerciements	8
--	---

PRÉFACE

Par Jeff Sutherland - Co-Créateur de Scrum

As Scrum enters the mainstream, Xebia 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

INTRODUCTION

Ce chapitre explique la genèse et les principes fondateurs de la Scrum Master Academy. L'idée de départ m'est venue début 2012 alors que je cherchais un moyen de transmettre mon expérience agile à mes collègues de Xebia dans un format pédagogique, sans écrire un livre de 500 pages et sans donner de certification formelle.

J'ai acquis de nombreux réflexes agiles « sur le tas » en me confrontant aux difficultés de certaines situations et sans spécialement lire de livre. 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 ? J'ai 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. J'ai aussi remarqué que mes entorses à certaines méthodes agiles étaient parfois perçues comme des actes de crime contre l'agilité. J'ai 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 ré-inventer la roue ?

Comme souvent dans ce genre de situation, la réponse est là où on ne l'attend pas : sur mon canapé devant la télévision. Pour l'occasion, c'est la série américaine NCIS, diffusée le vendredi soir sur M6, qui m'a inspirée. 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.

Avec Jean-Laurent de Morlhon, nous avons travaillé d'arrache-pied (le mot est faible) pour rassembler nos expériences agiles depuis 10 ans et définir un ensemble de règles « à la NCIS ». Nous avons ensuite présenté une première mouture de la Scrum Master Academy à nos collè-

gues, habillés de treillis militaires, sifflet au bec et casquettes pour renforcer l'impact. Nous avons apprécié cette présentation, non pas parce que nous aimons nous habiller en treillis mais pour les deux raisons suivantes. Tout d'abord parce qu'elle a permis aux participants de partager leurs expériences, à travers nos règles, et de mettre sur la table des situations parfois délicates. Ensuite, parce que cela a fait naître, chez Jean-Laurent et moi même, quelque chose qui nous manquait : la volonté de formaliser nos expériences en confrontant nos histoires, sans tomber dans le syndrome de la brève de comptoir. Le résultat auquel nous sommes arrivés dépasse largement l'idée initiale.

C'est donc tout naturellement que nous avons proposé cette session pour le Scrum Day 2012. Pour l'occasion, nous n'avons pas présenté les 50 règles car le créneau d'une heure ne suffisait pas. De plus, le contenu complet nécessite un minimum d'échanges avec les participants. L'exercice s'annonçait délicat car le propre de l'agilité est de ne pas donner de règles d'or et nous allions précisément faire le contraire devant un auditoire comprenant des agilistes forcenés.

Le plébiscite que nous avons reçu après cette session publique nous a encouragé à poursuivre le chemin et à formaliser les règles de manière un peu plus détaillée. Je vais donc vous dévoiler dans ce livre les 50 règles de la Scrum Master Academy. Chacune sera accompagnée de quelques explications.

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. Dans un passé lointain, où j'étais ingénieur qualité, on parlait beaucoup de deux niveaux de qualité : la qualité du produit et la qualité du processus. Le Scrum Master est garant de la qualité du processus, même s'il n'en est pas le dépositaire unique. Nous avons donc adressé cette liste de règles au garant de la qualité du processus.

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

Parlons de valeurs

Fin 2012, l'agilité était en France une réalité pour de nombreuses équipes de développement. Beaucoup d'observateurs s'accordent à dire que 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, [l'enquête nationale¹](#) qu'avait menée le French Scrum User Group en 2009 montrait déjà l'ampleur du phénomène. Les enquêtes régulières, menées par l'éditeur de logiciel VersionOne, nous éclairent également sur la profondeur de cette adoption au niveau mondial. Les retours d'expérience documentés sont nombreux, pas seulement sur de petits projets, mais également sur de gros programmes, n'en déplaise aux détracteurs de l'agilité.

Cette popularisation entraîne inévitablement une dilution des valeurs et des principes fondateurs parmi la masse des adoptions. Les agilités 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 ne pas rater le vent nouveau qui souffle sur le développement logiciel, mais qui ne veulent pas vraiment changer de culture. Ce chapitre vous présente un ensemble de règles pour ne pas perdre de vue l'essence de l'agilité.

¹- <http://bit.ly/18OnUm2>

Règle n°1 : CONNAIS TES VALEURS AGILES



J'ai choisi avec soin la règle n°1. En effet, c'est une place symbolique, il fallait une règle emblématique. La règle n°1 est là où tout commence, la base de la base, la règle dont on se souvient.

Nous commençons donc avec une règle pleine de bon sens et à la limite de l'évidence pour tout agiliste qui se respecte. Mais la réalité n'est hélas pas aussi triviale. Lors de notre session au Scrum Day 2012 avec Jean-Laurent, avant de dévoiler cette première règle, nous avons posé les questions suivantes à la cinquantaine de participants:

- Qui pratique l'agilité au quotidien ? Pouvez-vous vous lever ?
90% de la salle était debout.

- Qui connaît par coeur les 4 valeurs du Manifeste Agile ?
Si vous ne pouvez pas citer les 4 de mémoire asseyez-vous.
Il restait moins de 10 personnes debout.
- Qui connaît par coeur les 12 principes du Manifeste Agile ?
Les dernières personnes debout se sont assises.

Cette étude de marché complètement improvisée est malheureusement assez représentative de notre niveau de maîtrise [des valeurs de l'agilité](#)². Je m'inclus dans ce constat car je suis bien incapable de citer de mémoire les [12 principes](#)³ qui découlent des 4 valeurs. Il se trouve que le manifeste agile est plein de bonnes ressources. Les réponses à certains des problèmes que nous rencontrons sur le terrain se trouvent souvent dans ce manifeste.

Le manifeste agile nous pousse également à garder du discernement dans l'application d'une méthode agile. Tout agiliste fait des compromis avec la méthode qu'il applique, que ce soit Scrum, XP ou autre. Ce qui est important, c'est d'être cohérent avec les valeurs qui sous-tendent cette méthode et ces valeurs sont celles du manifeste. Par exemple, je vois sur le terrain des équipes se perdre dans des débats sur, par exemple, le manque de formalisme des critères d'acceptation, alors même que le Product Owner est un nouvel arrivant qui connaît à peine son produit. Voici un syndrome typique d'une équipe qui a remis l'accent sur le suivi d'un processus, certes agile, plutôt que de chercher si les individus sont les bons et s'ils ont les bonnes interactions entre eux.

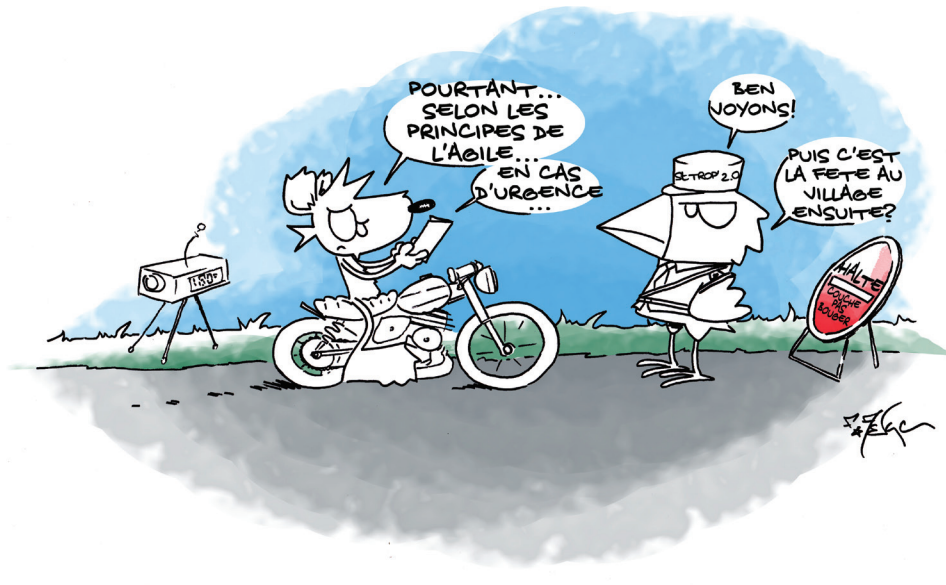
Autre exemple d'un principe souvent oublié: « Les gens du métier et les développeurs doivent travailler ensemble **chaque jour** durant l'ensemble du projet. » Est-ce le cas dans votre projet agile ?

Beaucoup des règles qui suivent dans notre liste ne seront qu'une pâle redite du manifeste. C'est pour cela que s'il ne doit rester qu'une règle, c'est la n°1. Relisez le manifeste et les 12 principes qui en découlent.

2- <http://bit.ly/GKO287>

3- <http://bit.ly/1gpx38T>

Règle n°2 : LE MANIFESTE AGILE N'EST PAS LA DÉCLARATION DES DROITS DU DÉVELOPPEUR



C'est une règle qui s'avère polémique à souhait et qui vient en complément de la règle n°1. Nous espérons bien qu'elle fera couler de l'encre sur Twitter.

Les strates décisionnelles ne sauraient être les seules responsables des échecs ou des succès en demi-teintes des initiatives agiles. J'observe au sein de certaines équipes agiles, l'émergence de comportements qui se réclament de l'agilité mais me paraissent dépasser largement le cadre du manifeste agile. Ces comportements peuvent prendre la forme de combat social « J'ai le droit de ne pas faire d'heures supplémentaires »

(même quand le projet est en souffrance) ou de lutte des classes « J'ai le droit de dire 'Non' à mon client » (quand je ne suis pas d'accord avec lui) ou encore de plaider pour la sauvegarde des libertés individuelles « J'ai le droit de choisir mes tâches » (surtout celles qui sont intéressantes).

Je force volontairement le trait, mais n'appelons pas « agiles » tous les arguments qui visent à améliorer le confort et le bien-être des équipes de développement. À l'heure de son adoption massive, cela me paraît desservir le mouvement agile plus que l'encourager.

Il faut reconnaître que le mouvement agile draine dans son sillage un écosystème riche de pratiques managériales et organisationnelles plus humaines. Ces pratiques tirent leurs inspirations de la psychologie sociale ou de l'observation de systèmes complexes. Les experts agiles piochent allègrement dans ce vivier pour étayer le bien-fondé des principes agiles ou développer de nouvelles approches, comme par exemple : [Radical Management](#)⁴, [Cynefin Model](#)⁵, [Management 3.0](#)⁶. Certains créent même de nouveaux mouvements à part entière, comme le réseau [STOOS](#)⁷. Cette démarche, louable pour des experts, ne doit pas faire oublier la nécessaire courbe d'apprentissage pour maîtriser ces concepts. Je conseille à un Scrum Master qui fait ses armes de rester simple dans son discours et son application des techniques agiles. Si vous venez juste d'apprendre à skier, vous n'allez pas prendre une piste noire (je l'ai fait mais j'étais jeune), vous risqueriez de vous blesser et de mettre les autres skieurs en danger. Alors je m'adresse à toi, jeune Scrum Master, ne te laisse pas griser par les perspectives d'un lendemain qui chante grâce à l'agilité. L'agilité met les problèmes en lumière beaucoup plus vite et les problèmes s'appellent souvent « course à la productivité », « manque de budget », « indisponibilité », « visions divergentes », « culture d'entreprise », « décisions des actionnaires », etc. Les réponses à tous ces problèmes sont à chercher ailleurs que dans le manifeste agile. Cette règle n°2 vise à apporter du discernement lors de l'utilisation du mot agile.

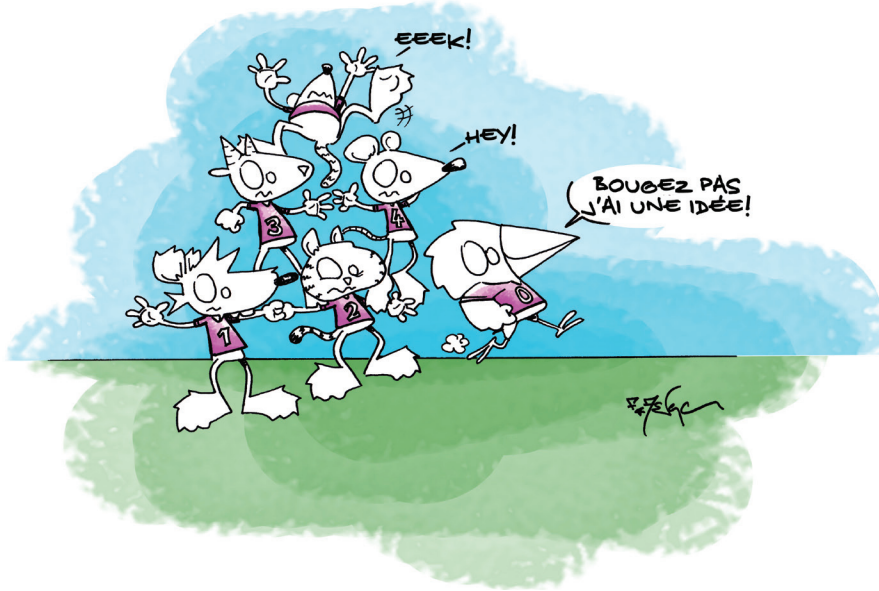
4- <http://bit.ly/19idpbx>

5- <http://bit.ly/1bYnyuv>

6- <http://bit.ly/19fPYOl>

7- <http://bit.ly/1a9LqrE>

Règle n°12 : ON TRAVAILLE TOUJOURS EN ÉQUIPE



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. 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. Je commence généralement mes interventions d'accompagnement en allant boire un verre avec toute l'équipe (donc y compris le Product Owner) à la fin du premier sprint, c'est bête mais ça forme des liens. On discute dans un

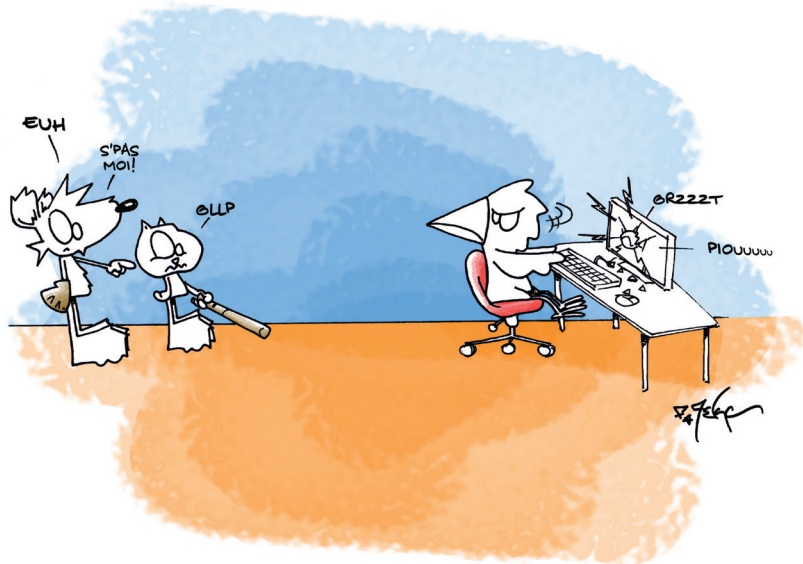
environnement différent et les informations ne sont pas les mêmes, on parle de nos enfants, de nos expériences similaires, on développe une sympathie pour l'autre, ce qui sera un atout non négligeable dans les coups durs. Avant même ce verre, j'organise un déjeuner commun si les membres de l'équipe n'ont pas l'habitude de se retrouver pour manger. Ce n'est pas tout, je constate souvent qu'une équipe qui démarre avec Scrum passe par un état d'euphorie lors de son premier sprint. Je vois souvent des post-its fleurir à la première rétrospective expliquant combien il est agréable de travailler avec cette équipe.

Il y a quelques années, je disais « Bravo » et nous passons ensuite aux problèmes pour essayer de trouver des points d'action. Je constate avec l'expérience que cette belle euphorie est fragile et qu'elle a tendance à voler en éclat avec les premières difficultés (c'est à dire lors du sprint 2).

Aujourd'hui, je passe un peu de temps à la première rétrospective pour creuser les raisons de cette euphorie : pourquoi avez-vous apprécié travailler avec cette équipe ? Je recueille généralement quelques regards vides ou des réponses triviales du type « parce que c'est bien ». Il faut généralement 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.

Je note soigneusement ces éléments sur des post-its que j'affiche ostensiblement dans le bureau, ce sera la «Charte d'Équipe» pour les sprints à venir.

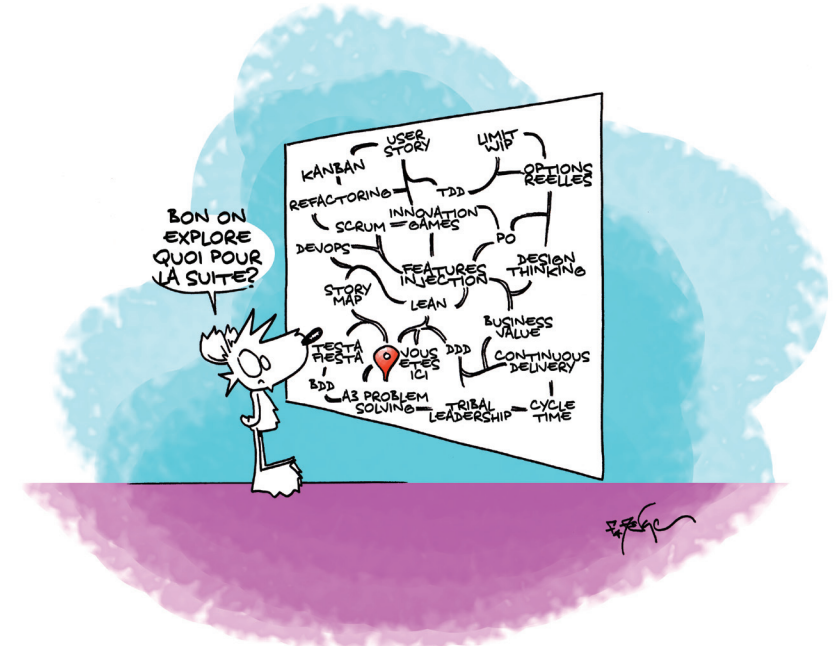
Règle n°19 : NO BROKEN WINDOW



La théorie de la fenêtre cassée est issue d'expérimentations sociales datant du milieu du XX^e 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. 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 traînent.

Règle n°21 : EVALUE 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. L'agilité est un chemin 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.

Le plus simple est le site [A better team](https://www.abetterteam.com/)⁸ qui propose un questionnaire appelant des réponses binaires et, à l'issue de la saisie, un niveau de sécurité rouge, orange, ou vert. Vous allez voir qu'il est difficile d'obtenir

8- <http://bit.ly/1bYobnK>

le niveau vert. Le formulaire est basé sur les écrits de James Shore, très orienté eXtreme Programming. Vous avez aussi une checklist plus orientée Scrum mise au point par Henrik Kniberg et utilisée par Jeff Sutherland lui-même, ou encore un questionnaire d'enquête mis au point par Dean Leffingwell. 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.

Parlons des user stories

Parlons des user stories

Les 5 règles de ce chapitre se focalisent sur un élément qui s'avère paradoxalement souvent mal maîtrisé lors d'une réalisation d'un projet agile : les user stories. Ce n'est pas à proprement parler un élément du framework 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 j'ai pu rencontrer :

- faut-il exprimer un besoin ou une solution ? Je 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.* »

J'aime la définition suivante (qui n'est pas de moi bien sûr) :

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 spécificités se retrouvent dans les caractéristiques des 3C⁹ ou l'acronyme INVEST¹⁰ 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 4 règles suivantes, qui globalement tendent toutes vers le même objectif : une bonne collaboration entre l'équipe technique et le donneur d'ordre.

9- <http://bit.ly/1cupiye>

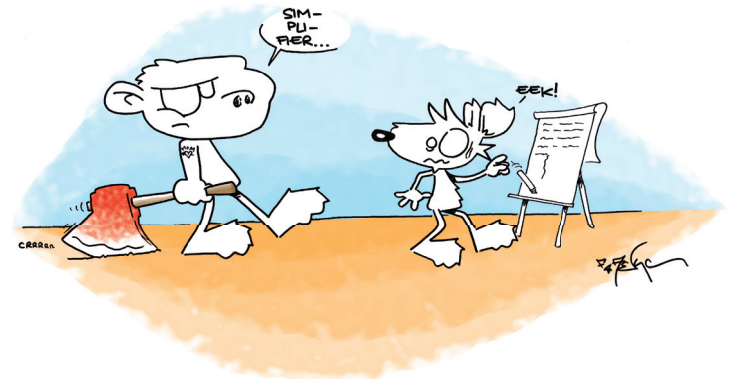
10- <http://bit.ly/16QWleK>

Règle n°4 : N'OUBLIE JAMAIS LA VALEUR D'UNE STORY



Cette règle vise 3 objectifs pour notre 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'inversement la valeur est très difficile à estimer en argent sonnante et trébuchante. Dans les 2 cas, on peut appliquer quelques trucs 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... 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 termes de «risque de ne pas faire». Cette dernière s'applique souvent pour les stories qui découlent d'obligations réglementaires, mais on peut l'étendre à n'importe quel type de stories.

Règle n°5 : UNE STORY TROP GROSSE POUR ENTRER DANS UN SPRINT EST UNE EPIC

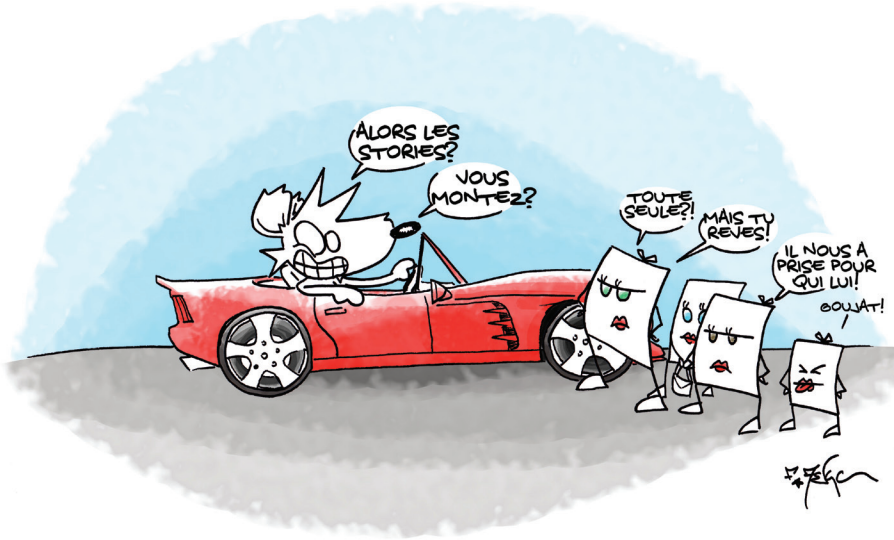


Cette règle est un simple rappel de ce que l'on enseigne classiquement dans un cours Scrum ou une formation agile. Il nous paraît cependant nécessaire de le rappeler car ce n'est pas forcément toujours bien appliqué. On pourrait, en fait, reformuler cette règle de la manière suivante : si la story est 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 ne peut pas être traitée comme telle par l'équipe de développement, il faut la découper en unités plus petites qui vont rentrer totalement dans un sprint. Ici, je me retrouve parfois confronté à la situation où le PO, ou l'équipe, considère qu'on ne peut pas découper plus finement certaines stories et découpent donc la story en tâches techniques planifiées sur 2 sprints. Pourtant, il existe de nombreuses [stratégies](#)¹¹², 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).

11- <http://bit.ly/GMyvVD>

12- <http://bit.ly/1gpynsj>

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, dont la paternité revient à [Jean-Laurent](http://bit.ly/1gpyqEF)¹³. 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. J'irai 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. Jean-Laurent l'explique de la manière suivante : les estimations c'est un peu le jeu des devinettes. Nous savons qu'elles sont globalement vraies mais précisé-

13- <http://bit.ly/1gpyqEF>

ment 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 1/4 de l'objectif 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.

Règle n°8 : À LA FIN DU SPRINT, IL Y A 2 OPTIONS POUR UNE STORY : « DONE » OU « NOT DONE »



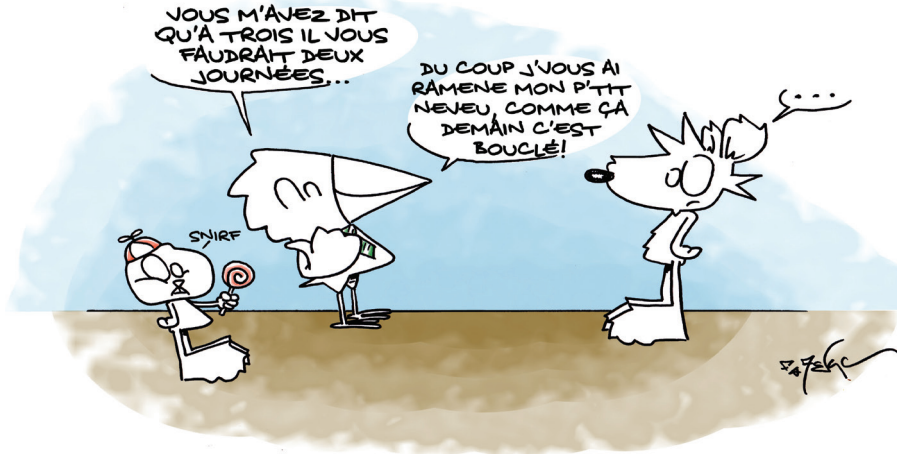
Il s'agit encore d'une règle évidente qui est enseignée dans de nombreuses formations agiles. Et pourtant, les tentations d'y déroger sont nombreuses : « il ne reste plus qu'à intégrer », « on y est presque », « on n'a pas testé mais j'ai confiance », « on n'a qu'à dire que c'est done et on ajoutera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

Voilà la manière dont je fonctionne : oui, je reporte totalement les points, quel que soit l'état d'avancement des stories concernées. Bien souvent, celles-ci avaient de toute façon été sous-estimées. Si l'on ne reporte pas les points, on risque de sous-estimer le contenu du sprint suivant et donc, de se retrouver à nouveau avec des stories non terminées. Et progressivement, sprint après sprint, on se construit un matelas de stories « not done » qui prend du volume. Très vite, la vélocité ne signifie plus rien, donc les projections ne signifient plus rien, donc le PO ne comprend plus l'avancement et tout ça risque de se terminer en un vaste règlement de comptes.

Pour résumer : je ne compte pas une story « not done » dans la vélocité constatée d'un sprint et je la compte entièrement dans les points prévus pour le sprint suivant, sachant que ce total de points 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.

Règle n°9 : TU N'ESTIMERAS PAS EN JOUR/HOMME



De l'efficacité des cérémonies Scrum

Utilisez les story points ou tout autre échelle relative pour vos estimations sur les éléments du Product Backlog. Evitez autant que faire se peut l'utilisation des jours-hommes pour les estimations, ils sont trop rapidement mal interprétés et amènent à des erreurs souvent bien plus importantes. Au delà des stories, évitez le chiffrage des tâches en heure, ce travail n'apporte que peu de valeur et dérape trop souvent dans la durée.

De l'efficacité des cérémonies Scrum

Dans ce chapitre, je vais revenir aux bases de l'activité de 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 toutes façon vous n'en avez plus), et votre tableau d'obstacles n'est plus à jour depuis longtemps. Votre priorité n'est plus de faire du logiciel 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 (à prononcer avec l'accent du Sud-Ouest).

Règle n°3 : LE BURNDOWN EST PUBLIÉ TOUS LES JOURS APRÈS LE STAND UP



Le burndown 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. J'imagine que les créateurs de Scrum ont voulu garder une bonne balance entre simplicité et transparence maximum. Paradoxalement, je constate trop souvent l'absence de burndown affiché sur le mur des équipes agiles. Quand je pose la question, les explications tombent souvent dans l'une des deux catégories suivantes.

« Le burndown est dans Jira (ou VersionOne, ou ...), il suffit d'aller le consulter »
Les outils électroniques facilitent et automatisent certaines tâches fastidieuses. Ils permettent aussi un meilleur partage des informations dans un mode distribué. Dernier avantage, ils contentent les amateurs

de traçabilité. En revanche, je trouve qu'ils ont tendance à dissimuler l'information car ils nécessitent l'accès à un ordinateur et des comptes utilisateurs. D'autre part, ils sont en général peu souples sur le calcul des indicateurs : pour obtenir un indicateur correct, il faut avoir saisi toutes les tâches et toutes les estimations depuis le premier jour et certains n'aiment pas trop les ajustements : malheur à vous si vous ajoutez une tâche en milieu de sprint ou si vous retirez des stories, le résultat du tracé peut être... aléatoire. L'utilisation d'un outil électronique n'est pas une excuse pour ne pas afficher le burndown. Il suffit d'imprimer le graphique tous les jours et le scotcher bien en vue. On peut également préférer une technique moins gourmande en papier en traçant quotidiennement la courbe au crayon sur une seule feuille de papier (c'est d'ailleurs la façon dont il était utilisée initialement). Si la courbe n'est pas correctement tracée par l'outil choisi, cela prend 5 minutes de faire une version manuelle.

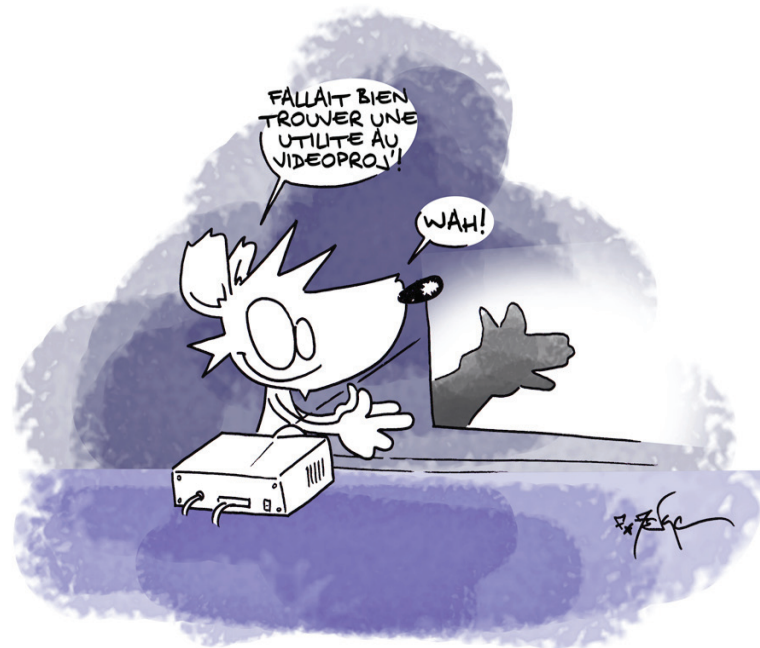
« Ça ne sert à rien, on sait où on en est »

Deux gros problèmes avec cette explication, le premier c'est d'assumer que le burndown ne sert qu'à l'équipe de réalisation. 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 ou tout autre interlocuteur s'intéressant au déroulement de la réalisation. Il participe à instaurer une confiance entre l'équipe et son environnement. Si vous êtes Scrum Master et pensez qu'on ne viendra pas vous déranger pendant le sprint pour savoir où vous en êtes, vous sous-estimez la curiosité et vous surestimez la patience de votre Product Owner et de vos dirigeants.

Il se trouve parfois que cette deuxième explication n'est pas le fait d'une confiance excessive de l'équipe mais un symptôme visible d'un mal plus profond, comme, par exemple, un manque de discipline sur l'estimation en tâches ou en points. C'est le deuxième problème que j'ai avec cette excuse, le burndown ne joue pas son rôle de révélateur du déroulement du sprint. Une absence de burndown peut même être caractéristique d'une organisation défaillante. En rétrospective, il est utile pour visualiser et analyser ce qui s'est passé. Le fait même d'avoir en permanence son problème sous les yeux met une pression plus forte pour saisir le problème à bras le corps.

Il existe bien d'autres mauvaises-bonnes raisons pour ne pas publier son burndown, vous pouvez les partager avec nous. Scrum Master, cette petite discipline quotidienne ne te veut que du bien et ne coûte pas cher.

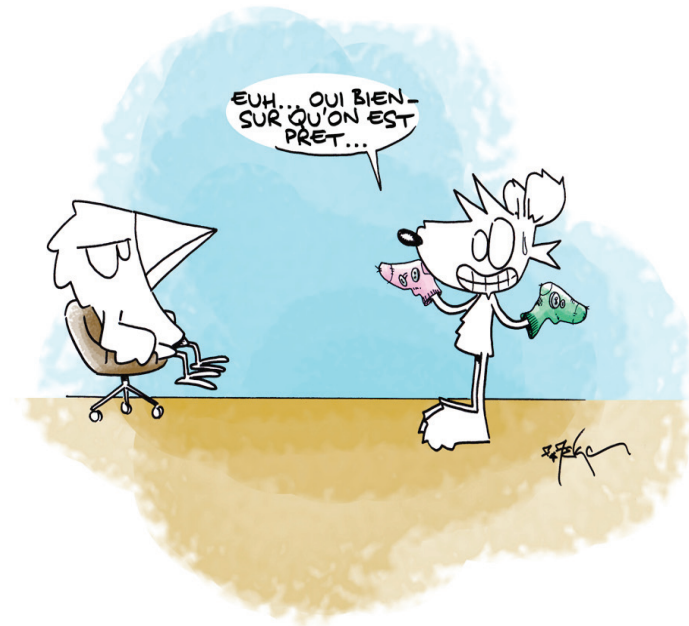
Règle n°30 : DÉMOS SANS SLIDES



La première dérive observée des cérémonies Scrum est souvent d'utiliser des présentations slides pendant la revue de sprint, dite démo. L'objectif de la démonstration est bien de montrer du logiciel opérationnel, bannissez alors les slides et préparez des scénarii d'utilisation réels avec les données adéquates. N'hésitez pas à vous inspirer des techniques de « [storytelling](#)¹⁴ » pour mettre en scène votre incrément de code. Souvenez-vous qu'une belle démo, fluide et bien organisée, laissera une impression positive et un sentiment de sérieux qui vous servira dans les coups durs.

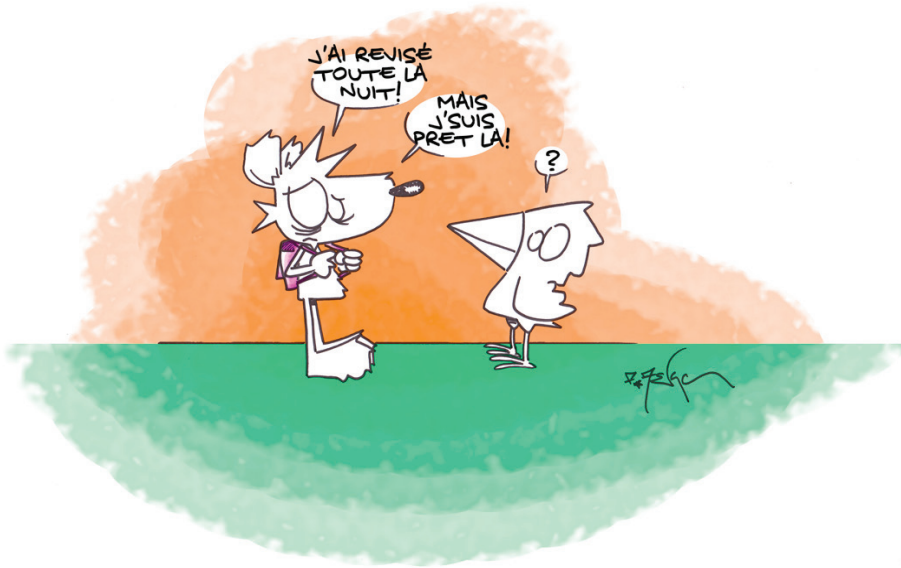
14- <http://bit.ly/15ZHND>

Règle n°31 : LA DÉMO COMMENCE À L'HEURE ET AVEC TOUT LE MATÉRIEL ADÉQUAT



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 est 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 aux participants, en leur indiquant que la démonstration commencera à l'heure. Rassemblez l'équipe 15 min avant la démo pour faire le point sur les rôles de chacun et revoir rapidement le contenu de la réunion. Branchez le matériel nécessaire et lancez les environnements. 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 une réunion 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.

Règle n°15 : EN RÉTROSPECTIVE, VARIER LES FORMATS



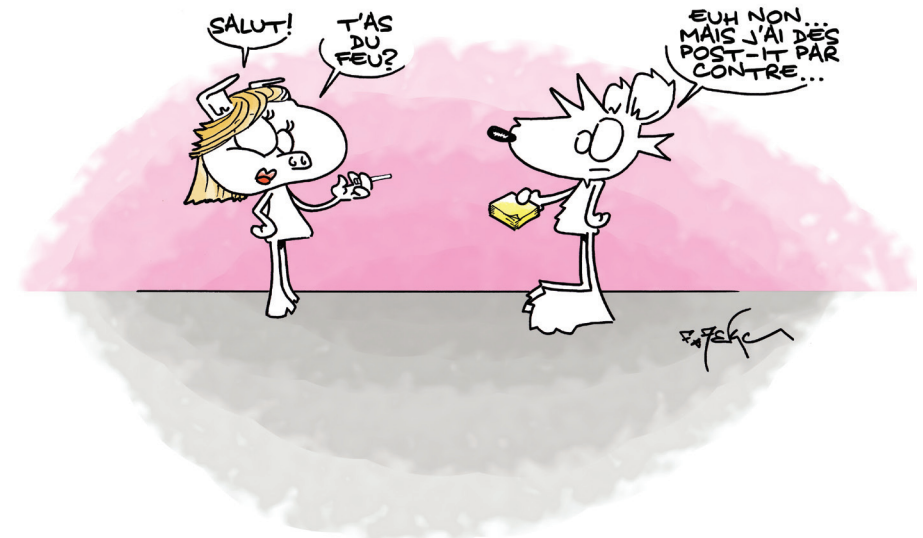
Parce qu'un format unique et systématique peut limiter la façon d'aborder les problèmes, varier de temps en temps les formats permettra de changer de perspective et renouvellera l'énergie de cette réunion. Évitez quand même de jouer avec tous les formats possibles et imaginables, vous n'êtes pas en train 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. Restez simple et n'essayez pas de résoudre la faim dans le monde.

Règle n°36 : FAVORISE LES TIG POUR GÉRER LES RETARDS EN DAILY SCRUM



Certains Scrum Master donnent des gages lorsqu'un membre de l'équipe enfreint une règle de comportement. Ces gages sont le plus souvent utilisés pour gérer les retards au Daily Scrum. Je ne vois pas d'inconvénient majeur à cette pratique, en revanche il faut rester cohérent. Par exemple, si vous imposez de donner 1 euro par retard, n'utilisez pas cette cagnotte pour aller boire un verre avec l'équipe, cela n'est pas vraiment dissuasif pour les retards. Préférez le versement de la cagnotte à votre manager, cela aura tout de suite un effet plus marqué. En ce qui me concerne, je préfère distribuer des travaux d'intérêt généraux (TIG) pour gérer les retards : s'occuper du build, écrire les comptes-rendus de réunions, s'occuper des commandes de post-its, ranger le bureau, ou vérifier la mise à jour de l'affichage visuel. Cela permet de délester un peu le Scrum Master de ses basses besognes néanmoins nécessaires et de sensibiliser tout le monde à leur intérêt.

Règle n°25 : TOUJOURS AVOIR DES POST-ITS SUR SOI



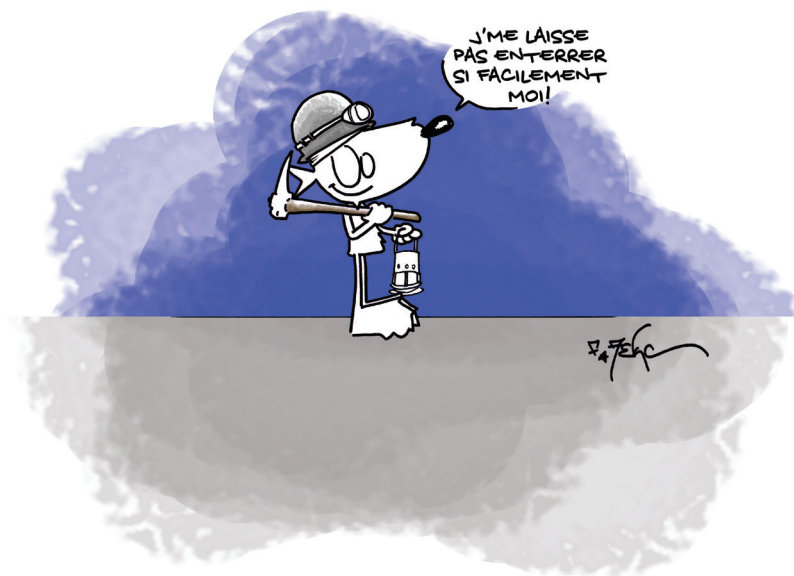
Outil emblématique de l'agilité, il n'est pas seulement un attribut folklorique du Scrum Master mais sert en toute occasion (planning, réunion de travail, retrospective, etc.). Il existe de nombreuses formes et couleurs pour diversifier votre utilisation, éviter de vous cantonner aux post-its standards. J'ai découvert cette année les post-its électrostatiques Statys grâce à @elpedromajor. J'ai découvert il y a quelques années les post-its flèches grâce à @jcgrosjean. Vous pouvez même faire imprimer vos propres post-its à votre goût et selon votre gabarit. Prenez les commandes avant de vous retrouver nu !

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 de 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 est inspirée du livre « *J'Arrête De Râler*¹⁵ » 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

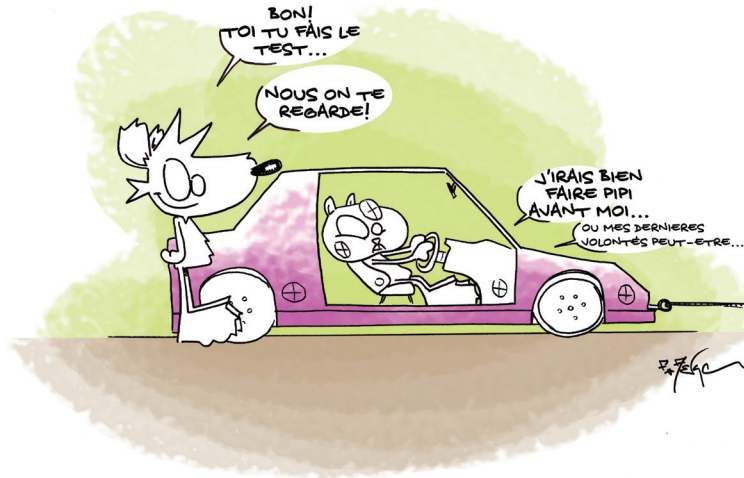
15- <http://bit.ly/1fi1UVR>

vous éparpiller et d'essayer de tout résoudre. Vous pouvez par exemple utiliser la [matrice de Merrill-Covey](#)¹⁶ 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.

16- <http://bit.ly/1fi1UVR>

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 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».

Règle n°22 : CONNAIS TON ENNEMI



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. Cherchez activement à identifier vos ennemis 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 vos meilleurs ennemis 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.

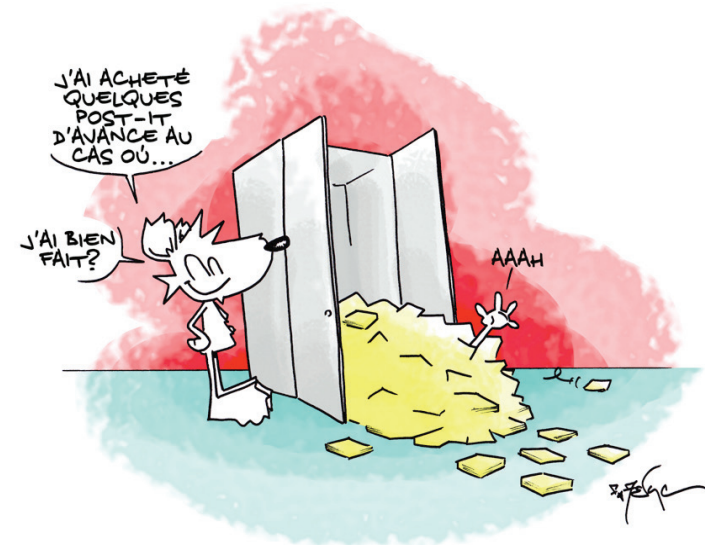
Règle n°26 : NE RIEN CONSIDÉRER COMME ALLANT DE SOI



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](http://bit.ly/1bYpMtX)¹⁷ 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 vous permettre de réévaluer les solutions envisageables sous un nouvel angle, vous découvrirez peut-être aussi que vous vous êtes autocensuré sur certaines solutions. Apprenez à changer les règles du jeu pour gagner en efficacité.

17- <http://bit.ly/1bYpMtX>

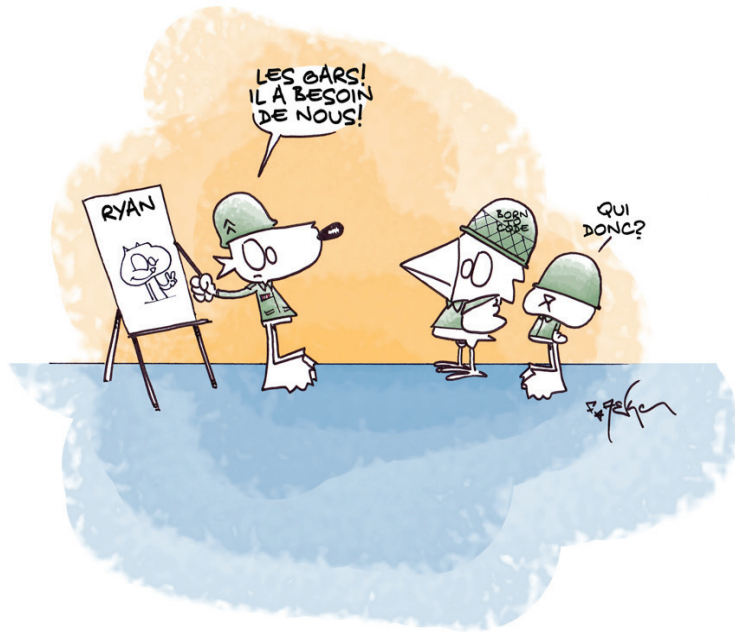
Règle n°27 : IL VAUT MIEUX DEMANDER PARDON QU'OBTENIR LA PERMISSION



Cette règle fait également partie des règles de Gibbs dont je vous parlais dans le chapitre d'introduction. Mais son origine nous vient d'une 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» Merci François¹⁸ pour la référence, je regarde trop la télé et pas assez les livres. Parce que les sprints sont courts et que l'on peut rapidement se retrouver bloqué par de nombreuses questions ou autorisations en attentes 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.

18- <http://bit.ly/1cuqXEd>

Règle n°35 : UN SCRUM MASTER SUR LE CARREAU EST INUTILE

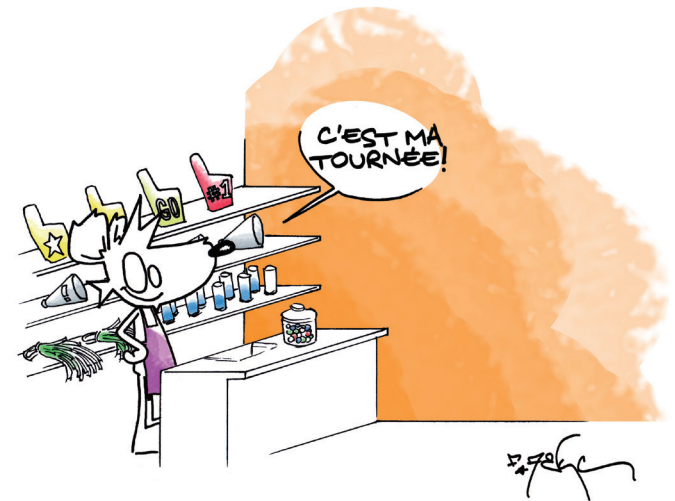


Corollaire de la règle n°27, votre pro-activité à gérer les obstacles ne doit pas vous conduire à votre mort professionnelle. Agissez dans les limites de ce que vous autorise la loi, la bienséance, le respect des individus, et les politiques internes de sécurité. Exploitez en revanche les zones de flou 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](#)¹⁹, [Daniel Pink](#)²⁰ 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](#)²¹:

19- <http://amzn.to/15ZlcWw>

20- <http://bit.ly/19zaebA>

21- <http://bit.ly/GKO287>

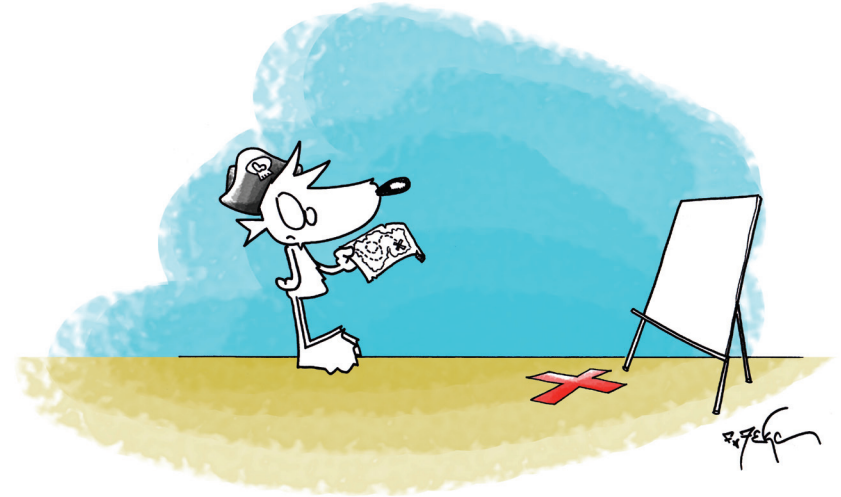
- **Autonomie** : « Réalisez les projets avec des personnes motivées. Fournissez-leur l'environnement et le soutien dont ils ont besoin et faites-leur confiance pour atteindre les objectifs fixés » et « Les meilleures architectures, spécifications et conceptions émergent d'équipes auto-organisées ».
- **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. Je considère 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. Je ne vais 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 sur [Internet](http://bit.ly/19ikqcr)²², dans des [livres](http://bit.ly/15UPnEO)²³ 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.

22- <http://bit.ly/19ikqcr>

23- <http://bit.ly/15UPnEO>

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.

Cette règle sous-entend également que vous fassiez usage du paperboard ou du tableau blanc à côté duquel vous êtes placé. 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. Je trouve que l'énergie est toute autre lorsqu'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. 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 [Gamertorming](#)²⁴, [Visual Meetings](#)²⁵ ou de [Bikablo](#)²⁶.

24- <http://bit.ly/1aoMlFX>

25- <http://amzn.to/19nkjFS>

26- <http://bit.ly/17Z9eB9>

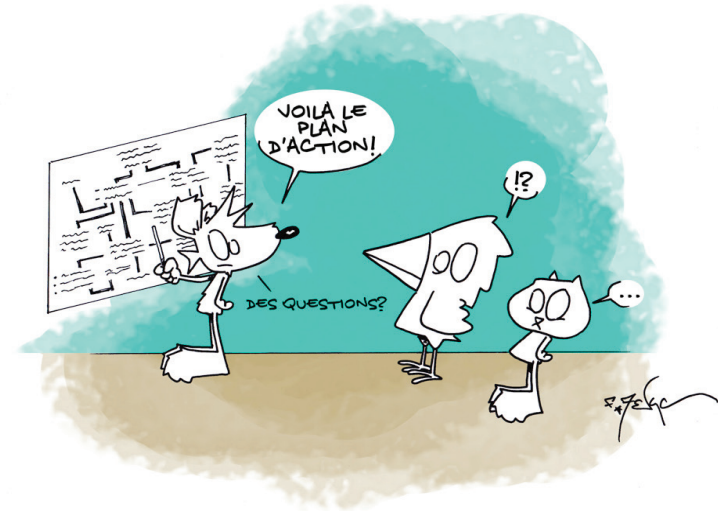
Règle n°18 : L'AUTO-ORGANISATION NÉCESSITE DU LEADERSHIP



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. Je trouve 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 projets 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). Je vois souvent 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. J'attends personnellement d'un Scrum Master qu'il ne soit pas seulement un super-administrateur des cérémonies 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 vélocité décline, les relations sont tendues, untel ne travaille pas en équipe, etc.

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. Il n'est pas question que le Scrum Master 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 exChefdeProjet) 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



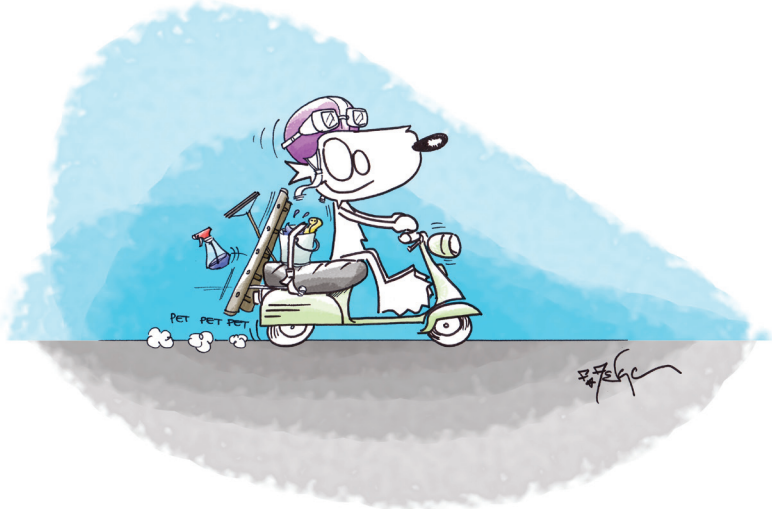
Une formation Scrum Master ne saurait donner toutes les informations pour réussir dans votre métier. Lisez des livres, des blogs (celui de Xebia est un très bon choix ;)), regardez des vidéos, venez participer aux conférences agiles, échangez avec vos pairs, créez des groupes d'intérêt dans votre organisation. En résumé, ne vous contentez pas du contenu du cours et de votre expérience.

Règle n°32 : ÉVITE LE BLÂME ET LA COMPLAISANCE



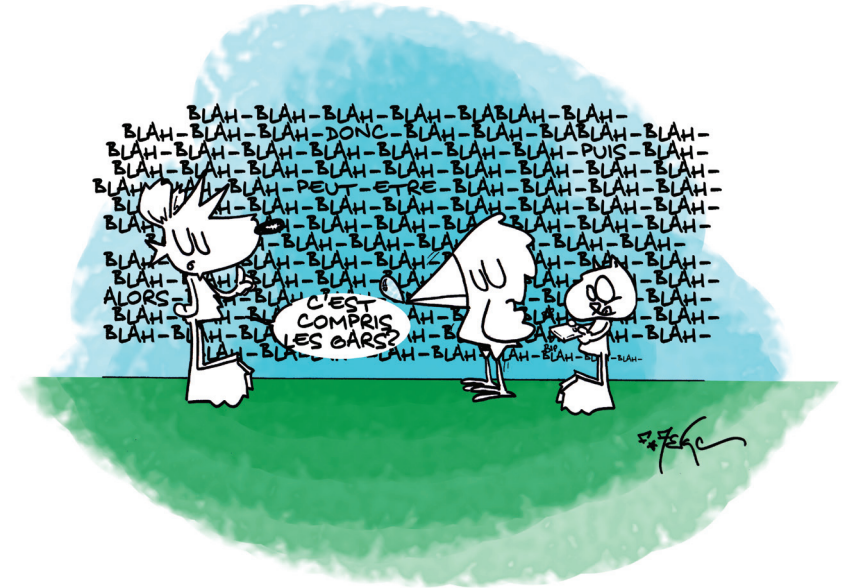
Corollaire de la règle n°19, cette règle est souvent appliquée dans les techniques de négociation raisonnée : il s'agit d'être dur avec la situation mais doux avec les personnes. Le Scrum Master a une position délicate par rapport aux autres membres de l'équipe: ce n'est pas leur supérieur, mais ce n'est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

Règle n°23 : ÉCOUTE ET TAIS TOI

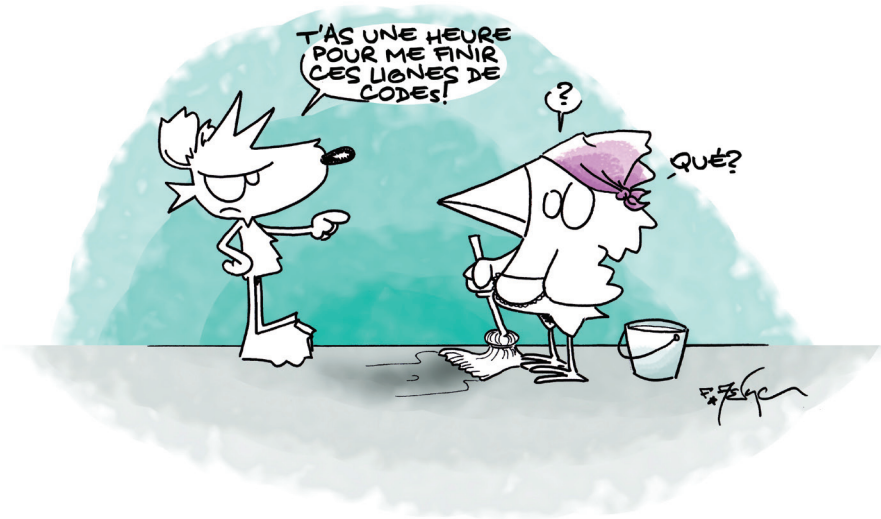


Quand on vous expose un problème, ne sautez pas trop vite aux conclusions et prenez le temps d'écouter jusqu'au bout votre interlocuteur. Laissez un temps de silence lorsqu'il a fini et reformulez ses propos. Il est peut être en train de vous donner la solution. Cette règle s'applique bien sûr en rétrospective, mais également en toute occasion : rencontre en tête à tête avec les membres de l'équipe, réunion de travail, discussions informelles, etc.

*Aller
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 oeuvre et qui nécessitent quelques trophées sur le tableau de chasse de votre expérience.

Règle n°24 : VEILLE À LA PLURALITÉ DES COMPÉTENCES DE L'ÉQUIPE



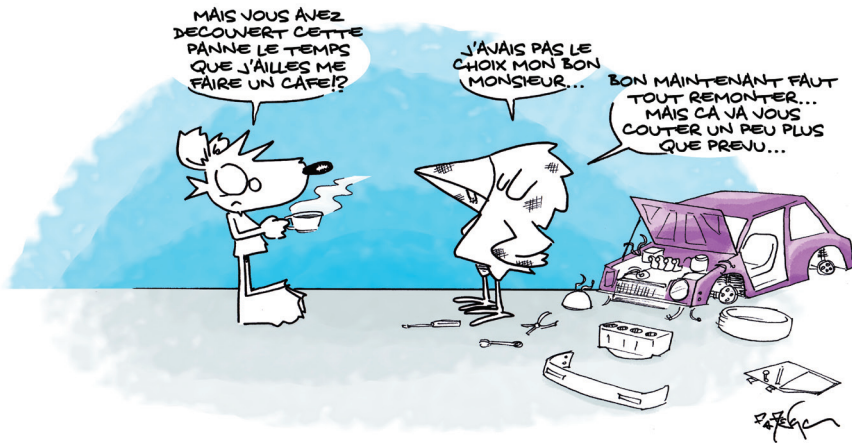
Pour assurer un travail en autonomie et livrer un incrément opérationnel du produit final, l'équipe Scrum doit être pluridisciplinaire. Le Scrum Master doit veiller à évaluer régulièrement que les compétences de l'équipe couvrent les compétences requises pour livrer du logiciel opérationnel en fin de sprint. Dans le cas contraire, ajoutez un obstacle à votre liste et agissez dessus.

Règle n°28 : NE CROYEZ JAMAIS CE QU'ON VOUS DIT, VÉRIFIEZ TOUJOURS



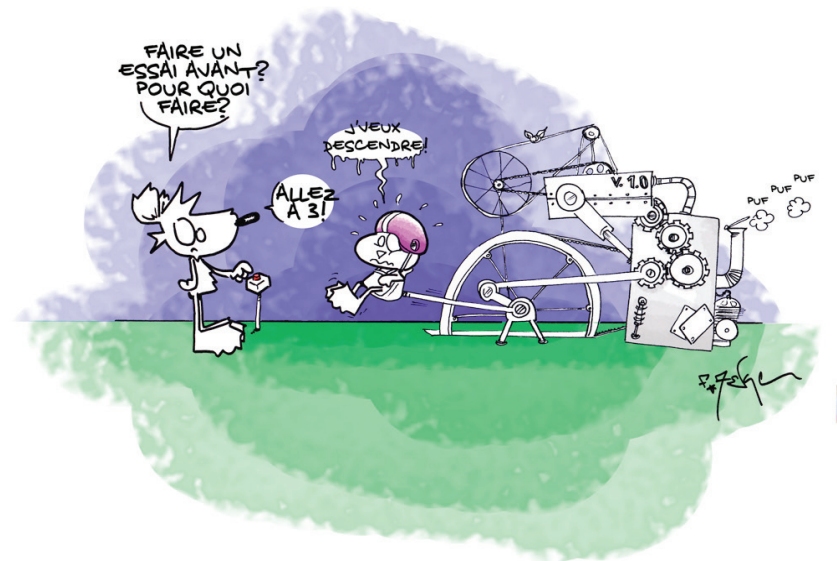
Cette règle s'applique en toute circonstance et permet de dépassionner les débats. Assurez-vous de passer du monde des opinions au monde des faits. C'est particulièrement vrai pour le travail de rétrospective, mais n'importe quelle discussion est sujette à cette règle.

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 et «indispensables au risque de provoquer une catastrophe nucléaire». Personne apprécie ne pas comprendre où passe son argent. Scrum nous vient des US, même si ses racines sont japonaises. J'ai donné cette année quelques cours certifiants avec un CST américain et j'ai été frappé de voir l'angle très financier qu'il donnait à sa formation. Pour les stagiaires peu habitués à manier les chiffres, certaines questions ou exercices étaient tout à fait déroutants : « combien coûte une équipe de 6 ETP sur un an ? » personne n'était capable de répondre, moi y compris (enfin le premier cours, le deuxième, j'avais la réponse). Et pourtant Scrum, ce n'est que de l'argent: le PO investit dans un sprint et obtient une valeur tangible à la fin du sprint. Si vous voulez être un bon Scrum Master, vous devez maîtriser les aspects financiers pour vendre les plans d'action sur la dette technique ou sur des opérations de maintenance à votre PO.

Règle n°33 : DÉPLOIE TOUS LES JOURS EN ENVIRONNEMENT DE TEST

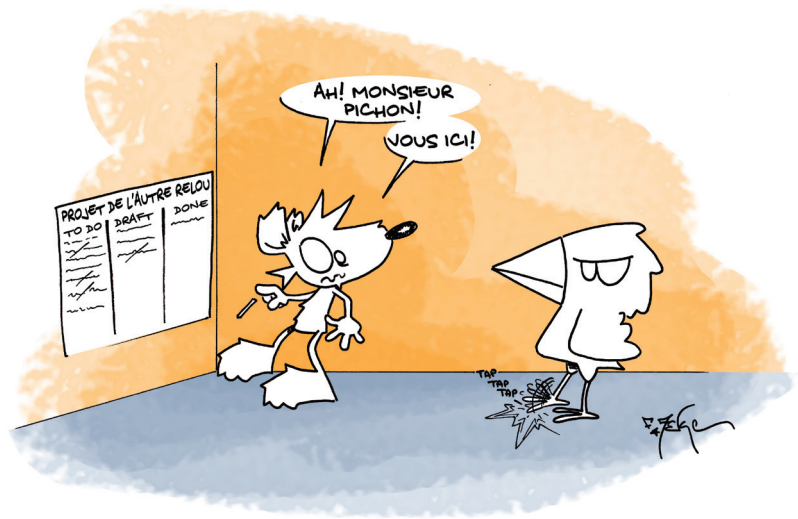


La seule mesure d'avancement en agile est le logiciel opérationnel. Avoir la discipline de livrer régulièrement en test vous poussera à renforcer vos pratiques de tests et lever les problèmes au plus tôt. Livrer régulièrement en production vous fera mettre la barre encore plus haut en termes de qualité et vous obligera à considérer tous les aspects insoupçonnés de votre produit (déploiement, migration, configuration, sécurité, performance).

Règles de crise

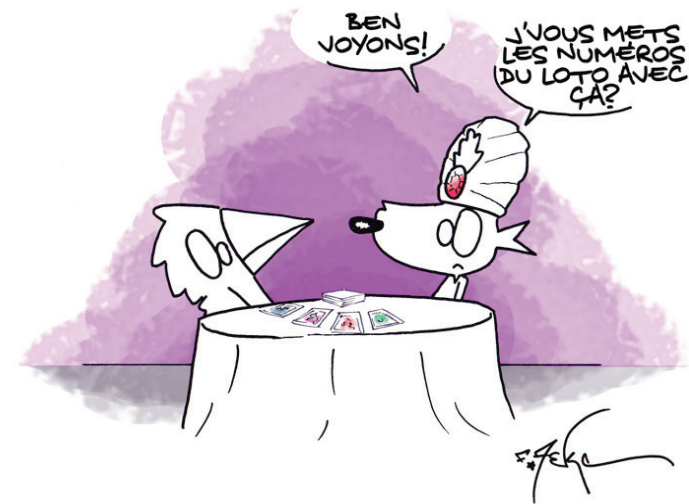
Les règles de crise ne sont à utiliser qu'en situation exceptionnelle. Leur objectif est d'amener un effet choc et de ne pas traîner pour prendre les décisions qui s'imposent. En effet, en situation de crise, plus vous traînez pour prendre des mesures radicales, plus la situation devient irréversible. 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, j'ai transformé une règle principale en règle de crise car elle ne méritait pas une attention continue.

Règle n°17 : TRANSPARENCE ET PAS INDÉCENCE



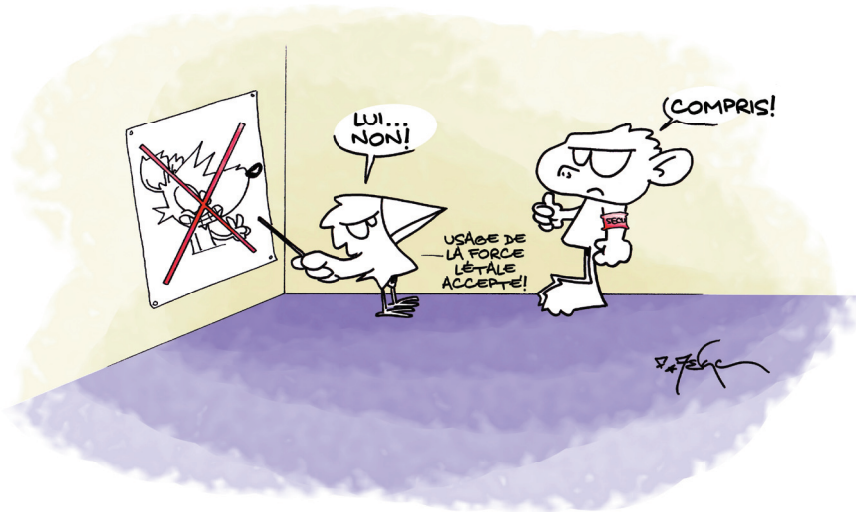
En situation de crise, assurez vous que les informations transmises par l'équipe sont formulées d'une façon qui ne porte pas offense au mandat de vos parties-prenantes. Parcourez votre affichage visuel avec les yeux de votre ennemi (cf règle 22) et assurez-vous que la formulation n'apporte pas une ambiguïté 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. Je me suis déjà personnellement fait épingler sur un post-it du tableau d'obstacles qui mentionnait «manque de maîtrise de la base de données». Quelques semaines après, à l'occasion d'une réunion de crise avec le DSI, j'essayais la réflexion que mon équipe ne maîtrisait pas la base de données et que c'était affiché ostensiblement. En réalité l'obstacle était le suivant : la base de données manquait de documentation et les nouveaux arrivants dans l'équipe ne maîtrisaient pas encore suffisamment le modèle.

Règle n°40 : LES POINTS DE STORY SONT UNE FOURCHETTE DE JOURS/HOMMES ET UN % DE CERTITUDE



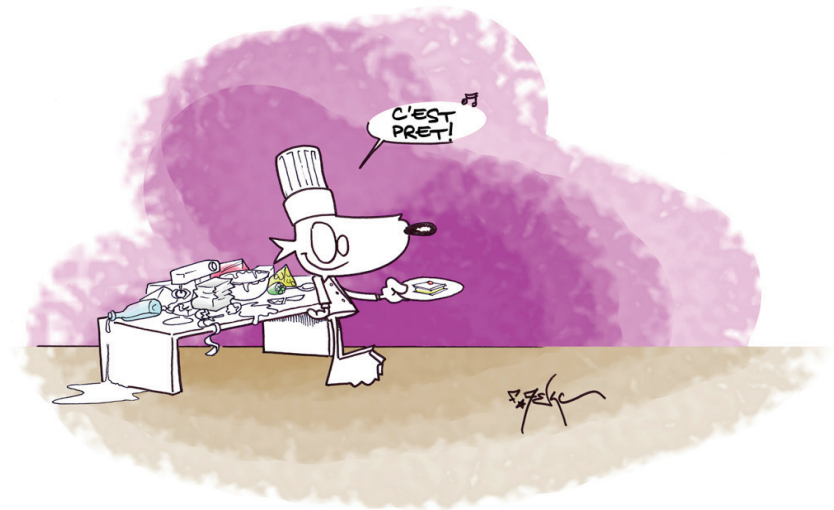
Dans les situations conflictuelles, il arrive que le story point ait une signification tellement obscure pour les néophytes qu'il contribue à maintenir un savant brouillard autour de la capacité de l'équipe. Ça c'est si vous faites bien votre travail, si au contraire vous êtes tombé dans le travers de compter des points partiels pour les stories non terminées et que vous finissez systématiquement vos sprints à moins de 80% de prédictibilité, c'est vous qui entretenez un flou artistique sur la capacité de l'équipe. Si vous êtes amenés à traduire vos estimations en jours/homme, utilisez toujours des fourchettes d'estimations accompagnées d'un pourcentage de certitude.

Règle n°41 : GÈRE LES POLLUEURS EN DÉMO



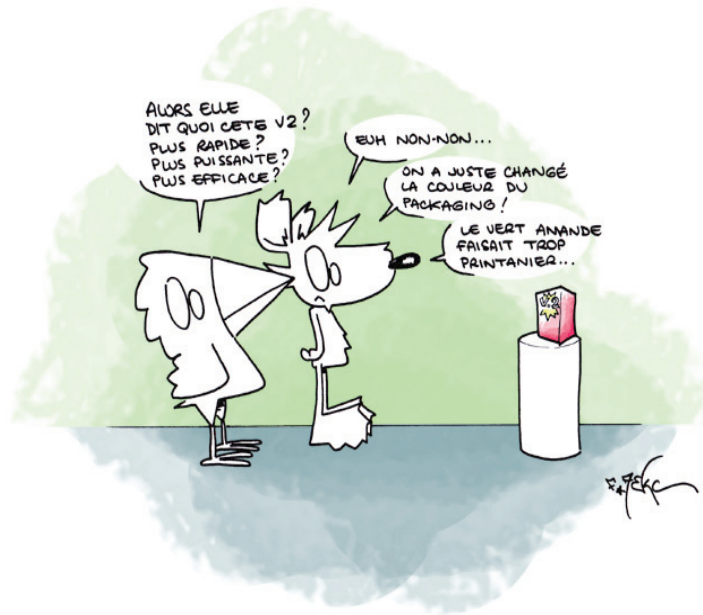
La revue de sprint est une étape importante de collaboration entre développeurs et business. Elle peut avoir un effet extrêmement positif sur les équipes, mais également se révéler très démotivante si vous ne prenez pas un minimum de précautions. Dans les situations de crise, n'autorisez que les personnes qui ont participé au sprint planning à venir à la démo pour ne pas dégrader la confiance entre le PO et l'équipe.

Règle n°42 : RANGE TON DÉSORDRE



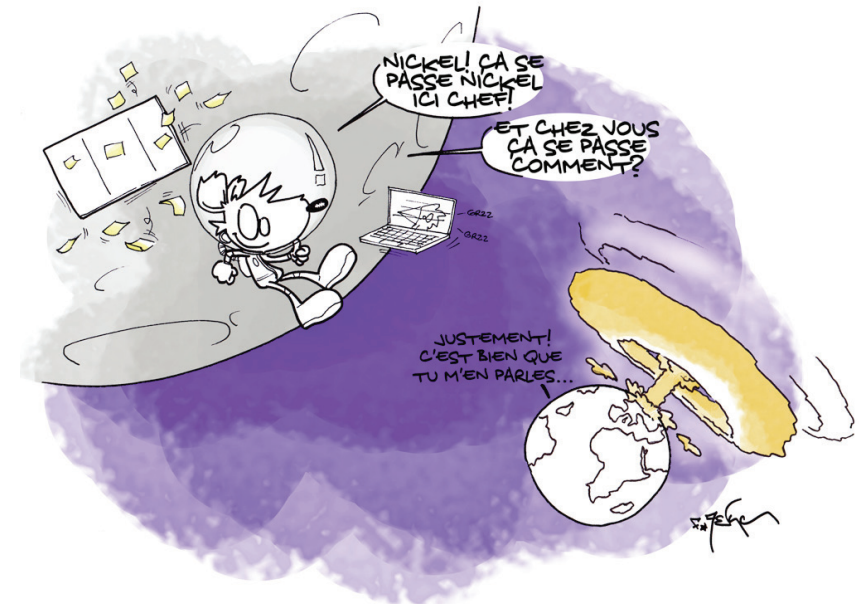
L'agilité et Scrum laissent une certaine marge de manœuvre à l'expérimentation et la gestion des échecs. En revanche, en situation de crise, il faut faire face à la réalité du business et parfois, faire l'effort nécessaire pour revenir à une situation propre : absorption de dette technique, livraison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

Règle n°43 : PAS DE SPRINT ENTIER DE REFACTORING



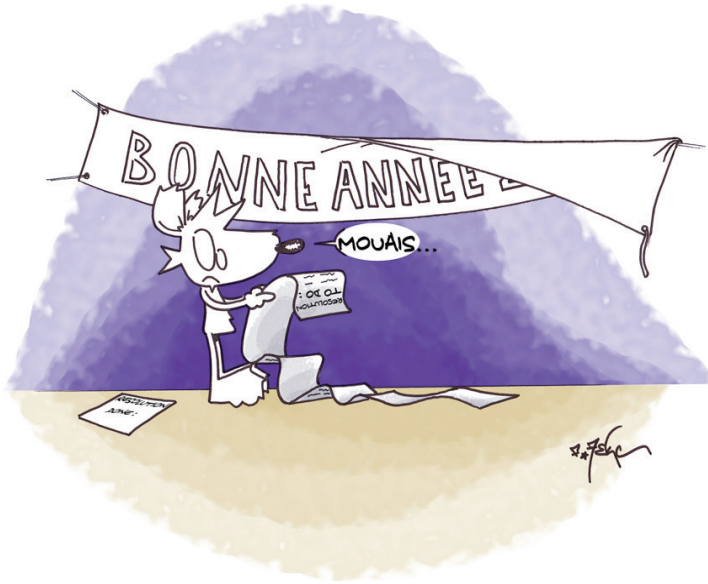
Les sprints entier de refactoring font passer un mauvais message auprès des donneurs d'ordre. En situation de crise, c'est l'aveu que vous avez perdu le contrôle de votre produit. Si votre dette technique est trop importante, essayez d'identifier au moins 2 sujets fonctionnels qui pourraient être adressés avec de la dette technique.

Règle n°44 : SI VOUS ÊTES EN SCRUM DISTRIBUÉ, TOUT LE MONDE SUR LE MÊME SITE



Il n'y a pas à hésiter une seule seconde, les objections financières ne sont que des détails à régler. L'argent que l'organisation perd avec un projet à la dérive vaut le prix des voyages pour mettre tout le monde ensemble. Il n'y a pas de limite à cette règle, si ce n'est l'atmosphère terrestre car cela va vous coûter un peu cher, mais l'offshore lunaire n'a pas encore été inventé. Si la dépense est jugée trop chère pour rassembler une équipe distribuée, c'est qu'il vaut mieux arrêter le projet.

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. Je fais 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 est 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 à 1h et prenez 2 actions d'amélioration sur les problèmes les plus graves.

Vos règles du jeu



Conclusion et remerciements

Chers Scrum Masters,

Vous avez lu évidemment ce livre depuis le début, et je vous en remercie. Si vous possédez un semblant d'esprit mathématique, vous avez dû remarquer qu'il manque 9 règles pour compléter la liste. Je vous en avais promis 50 et je n'en ai 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 46 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.

J'ai maintenant une bonne et une mauvaise nouvelle. La mauvaise est que ce livre est fini et que vous allez devoir lire les remerciements qui ne sont pas habituellement la partie la plus intéressante d'un livre.

Je voulais remercier Jean-Laurent de Morlhon, avec qui j'ai collaboré étroitement pour mettre au point cette liste au début de l'année 2012 et avec qui j'ai partagé des moments uniques pour les présenter d'abord aux Xebians en février 2012 et ensuite au Scrum Day 2012 fin mars.

Merci également à Florent, qui a consacré une partie de son temps extra-professionnel pour faire les illustrations et à Clément Rochas pour sa contribution aux idées d'illustrations.

Je souhaitais également remercier les Xebians qui ont relu mes billets de blog, corrigé mes fautes d'orthographe et posé des questions ennuyeuses. J'ai pris conscience de l'énorme travail d'écriture qui différencie une conférence et un livre avec les explications et les références pertinentes.

Mon dernier remerciement va à Luc Legardeur, Président de Xebia, qui m'a poussé à faire «quelque chose» pour partager mon expérience et sans qui j'aurais bêtement gardé tout ça pour moi ou mes proches collègues.

La bonne nouvelle, c'est que la Scrum Master Academy ne s'arrête pas à ce livre. Au delà des écrits, la formation Scrum Master Academy, qui figure au catalogue de Xebia Training, vous permettra de pratiquer chacune de ces règles dans un environnement sécurisé avec de nombreux exercices et mises en situation. Vous pouvez également télécharger un support pratique, du type «Cheat Sheet» pour garder ces règles sous la main en toute circonstance. Nous espérons par ces moyens professionnaliser un métier qui n'a pas atteint la maturité, car oui je l'ai appris après toutes ces années de pratique : Scrum Master est un métier, ce n'est pas un rôle.

Gilles Mantel

Xebia vous présente la Scrum Master Academy

Le succès de Scrum s'accompagne de la naissance de deux nouveaux métiers : Scrum Master et Product Owner.

Le nouveau paradigme managérial induit par les méthodes agiles remet profondément en cause certaines habitudes. Scrum est très facile à appréhender : 3 rôles, 3 artefacts et 3 cérémonies. Pour autant, la théorie ne suffit pas. Devenir un bon Scrum Master nécessite de l'expérience. Après la vague de certifications de ces dernières années, voici venue celle de la maturation.

Les Xebians ont accompagné plusieurs centaines de projets agiles dans le monde. Nous avons souhaité dans ce recueil, de manière parfois humoristique, vous faire partager nos bonnes habitudes.

À la manière d'un livre de cuisine, il vous laisse la place pour établir la liste de vos propres recettes tout en vous communiquant nos "tours de main". Nous espérons qu'il vous aidera à devenir un Scrum Master d'élite.

Bonne lecture

Luc Legardeur, Président fondateur de Xebia en France

A propos de l'auteur : Gille Mantel

Gilles est un consultant senior, directeur de projets agiles de plus de 10 ans d'expérience. Il est impliqué sur des projets itératifs depuis plus de 7 ans. Depuis 5 ans, il accompagne l'adoption des pratiques agiles et conseille sur les méthodologies pilotées par les tests (TDR, ATDD). Ses compétences sur les méthodes agiles et les tests sont reconnues et il participe régulièrement en tant qu'orateur à des conférences. Gilles est responsable de l'offre agile chez Xebia depuis plus de deux ans.

