



L'agilité, une extension du domaine du don

Claude Aubry, Jean-Pascal Boignard, Anthony Cassaigne

Version 1.00, 2020-11-29

Table des matières

1. Introduction	2
2. Don Lucas El PO	3
2.1. Son cheminement vers le don	3
2.2. Le paradigme du don	5
2.3. Le don dans les relations avec les utilisateurs	5
2.4. Le don dans les relations avec les parties prenantes	7
2.5. Le don pour ne pas ignorer l'impact sur le vivant	8
2.6. Le don au sein de l'équipe agile	10
3. Le don du point de vue du Scrum Master	13
3.1. L'adonnement improductif de Sarah	13
3.2. Sarah est accompagnée d'un coach agile	13
3.3. L'observation des relations dans l'équipe	13
3.4. La demande dans les relations de Sarah avec l'équipe	15
3.5. Le rite de la mêlée quotidienne pour raviver la dynamique du don	15
3.6. Le rite de la rétrospective pour rallumer le feu du don	16
3.7. Un cadre propice au don	17
3.8. La célébration dans le cycle du don	17
3.9. Le pardon mais pas l'oubli	18
4. Mob Programming & Don	19
4.1. Le mob pour apprendre	19
4.2. Le mob pour développer en équipe	19
4.3. Les rôles du mob et le cycle du don	20
4.4. Les valeurs du mob	21
4.5. Le mob pour amplifier la capacité d'écoute	22
4.6. Le mob et les choix de conception	23
4.7. Mob programming et rétrospectives	24
5. Conclusion	25

Remerciements

Nous tenons à remercier Étienne Appert et Patrice Courtiade pour le don de leurs superbes dessins.

Les dessins d'Étienne accompagnent le chapitre deux. Ils ont d'abord été créés pour le livre *L'art de devenir une équipe agile* (Dunod, 2019) pour illustrer l'exemple PermaBio.

Les dessins de Patrice sont une création originale.

Licence

Cette oeuvre est distribuée sous licence [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

Chapitre 1. Introduction

Cette période, l'avent, qui commence aujourd'hui, nous conduit à Noël.

Le fête de Noël est associée aux cadeaux dans l'imaginaire collectif. Quand vous faites un cadeau, vous pensez peut-être qu'il s'agit d'un geste désintéressé. Un don. C'est ce que nous pensions aussi avant de lire l'ouvrage choisi pour le klub de lecture de novembre 2020.

Ce livre, c'est **Extensions du domaine du don** d'Alain Caillé ; il nous entraîne à approfondir le paradigme du don, mis en évidence par le sociologue Marcel Mauss il y a 100 ans, et à l'étendre. Alain Caillé fait l'hypothèse qu'il régit l'essentiel de nos rapports aux autres. Après avoir lu son livre, nous sommes convaincus que l'agilité peut être considérée comme une de ces extensions.

Le cycle du don étendu, c'est : Demander - Donner - Recevoir - Rendre

Nous allons vous montrer — en vous faisant don de cet article — qu'on peut s'appuyer sur le paradigme du don étendu pour interpréter et consolider une approche agile. Et qu'en particulier, l'agilité **radicale** qui cherche à redéfinir la notion de valeur dans le développement de produits et services s'inscrit dans cette philosophie politique anti-utilitariste.

J'aurais presque tendance à dire que la clé de la rentabilité d'une entreprise, ou de son efficacité, c'est la vitesse de circulation des dons entre les salariés.

— Alain Caillé

En écho à cette citation de l'auteur, nous avons découvert que son concept de don s'applique à une équipe agile et à son écosystème : au sein d'une équipe, entre l'équipe et les parties prenantes, avec les utilisateurs. Il a également du sens pour interpréter les relations entre un coach et les coachés, dans une communauté d'agilistes et, bien sûr, cette initiative du calendrier de l'avent, ce christmas agile.

Recevez donc ce don (mais attention, cela vous fait entrer dans le cycle du don avec l'obligation de rendre !)

Nous allons décliner différentes facettes du don dans l'agilité avec trois points de vue :

- celui du Product Owner de PermaBio, don Lucas El PO
- celui de Sarah, Scrum Master, qui est en fait un point de vue bi-dons avec une réflexion sur le coaching et une autre sur la rétrospective,
- celui d'une équipe dans son avent-ure symbolique du mob programming.

Chapitre 2. Don Lucas El PO

Pour illustrer l'extension du domaine du don à l'agilité, commençons avec le point de vue de Lucas, le PO de PermaBio.



Figure 1. Lucas, Product Owner de PermaBio

2.1. Son cheminement vers le don

Avant de rejoindre PermaBio, Lucas travaillait comme analyste MOA dans une grande organisation. Il écrivait des spécifications à destination de l'équipe chargée du développement.

2.1.1. Sa découverte de l'agilité

Il a découvert l'agilité avec le manifeste ; le premier principe a été pour lui une révélation :

Notre plus haute priorité est de satisfaire le client en livrant rapidement et régulièrement des fonctionnalités à grande valeur ajoutée

— Manifeste agile, Premier principe

Bien sûr, satisfaire le client, tout le monde était pour, mais il sentait bien que le client était vite oublié dans les gros documents qu'il produisait et qui servaient de base de travail avec les développeurs.

2.1.2. Son rôle de PO

Le service où il travaillait devenait sensible aux méthodes agiles. Lucas, qui s'intéressait particulièrement à l'aspect produit a découvert la vidéo d'Henrik Kniberg^[1] sur le rôle de PO.

Une superbe animation de 15 minutes qui l'a enthousiasmé et lui a donné envie d'être Product Owner. Un PO vraiment dans l'équipe agile, capable de :

- prioriser en fonction de la valeur et de l'effort,
- dire non. NON.

Il a pensé que ce ne serait pas évident de pouvoir dire non dans sa grosse structure hiérarchisée, il a donc commencé par creuser les notions de valeur et d'effort.

2.1.3. L'effort

Lucas était content de voir que l'agilité se préoccupait de la notion de valeur. Dans les projets sur lesquels il avait travaillé, c'était uniquement le budget qui était considéré. On voulait savoir combien ça allait coûter. Et suivre le coût que prenait le développement rapporté au budget du début.

Il a découvert avec intérêt le **planning poker**. Tiens une pratique collective pour estimer la charge de travail, qui cherche à se dégager de la tyrannie des hommes jours en utilisant des points !

Il s'est dit que c'était une bonne technique pour quantifier le coût relatif des éléments du backlog. Même s'il n'était pas dupe des dérives possibles du planning poker, il était drôlement content d'avoir un chiffre pour chaque story et de suivre ainsi la vélocité de l'équipe.

2.1.4. La valeur

Si on avait l'équivalent pour la valeur, ce serait top. Cela permettrait d'ordonner le backlog de façon automatique en utilisant le calcul Valeur/Effort (le fameux ROI).

Seulement la valeur d'une story ce n'est pas simple à évaluer. Il a bien essayé le **priority poker** à la façon du planning poker, mais il s'est vite rendu compte que ça avait moins de sens sur des stories et que les développeurs n'étaient pas les bons interlocuteurs. Essayant avec des parties prenantes, il a constaté que les points de vue sur la valeur d'une fonctionnalité étaient très disparates ; de plus les utilisateurs étaient toujours bien loin des préoccupations.

La notion de valeur qui était considérée était uniquement de la valeur métier (ou business value). Dit autrement de la valeur économique. Dans le cas de Lucas, ce que ça allait rapporter aux actionnaires.

Or, de son point de vue, les priorités ne découlaient pas uniquement de cet aspect-là. D'autres notions étaient bien plus importantes pour Lucas : la satisfaction des utilisateurs, faire plaisir à une partie prenante, les dépendances entre les fonctionnalités, les risques soulevés par les développeurs...

Des choses non quantifiables.

2.1.5. Son épiphanie anti-utilitariste

Par ailleurs, sa participation à de nombreuses séances de planning poker et — surtout — l'usage pervers qui était fait de la vélocité par les managers l'a aussi amené à relativiser l'intérêt de l'estimation en points.

Lucas a eu la lucidité de reconnaître qu'il allait dans une impasse en essayant de se baser

uniquement sur des chiffres de valeur et d'effort pour prioriser. Adieu ses rêves de calcul pour ordonner son backlog !

Quand on est au coeur du développement de produits ou services comme Lucas, on est en relation avec des parties prenantes, avec des utilisateurs et bien sûr avec les développeurs de ces produits. Dans son rôle de PO, Lucas a pris conscience qu'il lui fallait prendre en compte le désir de reconnaissance de chacun plus que le seul intérêt rationnel économique.

Bref il était devenu anti-utilitariste.



L'anti-utilitarisme, c'est le constat qu'on ne peut pas tout expliquer par des intérêts économiques. L'affirmation qu'il y a d'autres notions de la valeur.

C'est ce à quoi s'attaque l'agilité radicale^[2] en l'élargissant ainsi :

Valeur = f (valeur métier, valeur d'apprentissage, impact sur le vivant, valeur sociale)

2.2. Le paradigme du don

Sans le savoir, Lucas était mûr pour appliquer le paradigme du don dans ses relations avec ses interlocuteurs.

Dans son **Essai sur le don**^[3], Mauss établit que les relations sociales sont basées sur une triple obligation : recevoir, donner et rendre. Il établit ainsi la base d'un mouvement anti-utilitariste.

Dans son livre **Extensions du domaine du don**^[4], Alain Caillé ajoute la demande — l'idée que le don répond à un besoin, ce qui nous va très bien pour l'appliquer à l'agilité — ce qui donne le cycle du don (DDRR) :

DEMANDER - DONNER - RECEVOIR - RENDRE

C'est à partir de ce paradigme du don étendu que nous allons explorer les relations de Lucas dans son écosystème^[5].

Commençons par les relations auxquelles il donne le plus d'importance, celles avec les utilisateurs.



L'exploration des relations de Lucas est réalisée à partir d'extraits de **L'art de devenir une équipe agile**^[6]. Le texte qui décrit les interactions dans l'exemple PermaBio est examiné à l'aune du cycle du don.

2.3. Le don dans les relations avec les utilisateurs

2.3.1. Lucas rencontre des utilisateurs

Lucas est donc le PO de PermaBio qui offre des services de fruits et légumes bio pour les

particuliers.

En fin de matinée, sur la recommandation de Victor, il part rencontrer des utilisateurs de PermaBio. Il va en profiter pour parler des nouveautés qu'il envisage d'inclure dans le produit (le ramassage des déchets pour le compostage et l'intégration dans mesrecettes.com) et demander leur avis aux participants. L'idée plaît bien, les participants réagissent positivement et on lui fait même une suggestion de collecte auprès des voisins.



Figure 2. Lucas sur le terrain

2.3.2. Le cycle symbolique du don basé sur la reconnaissance



Rappelons que le cycle c'est DRRR quand ça se passe bien. On le qualifie de symbolique parce qu'il rassemble.

Lucas est sensible à la qualité de ce qu'il mange. Il connaît pas mal de personnes de son entourage qui le sont aussi. Quand il priorise son backlog au nom de l'équipe dont il fait partie, il donne de son mieux pour faire plaisir aux utilisateurs du service. Il y a certes un aspect économique qui le guide, car le service est payant et fait vivre PermaBio, mais il y a aussi une dimension de reconnaissance. Il est plaisant à Lucas de dire aux autres ce qu'il fait, en quoi il participe à une action qui lui est chère ; il n'est pas indifférent à ses clients et utilisateurs, persuadé qu'il leur apporte plus qu'un accès à un service.

Lucas se donne dans son rôle de PO, non pas pour être mieux payé si le service se vend bien, mais pour cette reconnaissance.

Les utilisateurs, de leur côté, sont sensibles à l'intérêt que leur porte Lucas. Quand ils constatent que leurs idées sont prises en considération, par exemple s'ils la retrouvent dans le service proposé, ils éprouvent de la reconnaissance.



Le cycle du don n'est complet que lorsque les clients et utilisateurs **reçoivent** bien (par exemple en participant à la revue de sprint) et **rendent** (par exemple en donnant leur feedback à Lucas).

2.3.3. Les cycles diaboliques avec les utilisateurs

Le don, ce n'est pas toujours ce cycle symbolique — ou boucle de feedback — qui fonctionne bien du premier coup. De nombreux aléas peuvent mener au cycle diabolique IPRG (diabolique parce qu'il sépare) :

IGNORER - PRENDRE - REFUSER - GARDER

Voyons quelques situations typiques concernant le rôle de PO (le terme antipattern irait bien aussi pour les qualifier) :

- Le PO qui choisit une fonctionnalité alors qu'il n'y a pas de demande; il n'a pas le souci de l'attention de ce que veulent les utilisateurs.
- Le PO qui accepte une fonctionnalité qui va prendre des données aux utilisateurs sans leur demander leur avis.
- Les utilisateurs à qui l'on ne demande pas de donner leur feedback lors de la revue.
- Les utilisateurs qui gardent leurs ressentis pour eux.
- Le PO qui ignore le feedback.



On peut dire qu'une fonctionnalité ajoutée dans un produit n'est un don que si les utilisateurs la reçoivent bien, peuvent rendre par du feedback ET que si elle ne prend rien sans l'avoir demandé (par exemple des données personnelles).

2.4. Le don dans les relations avec les parties prenantes

2.4.1. Lucas rencontre des parties prenantes



Figure 3. Rencontre de Lucas avec les fondateurs de PermaBio

Cet après-midi, Lucas voit Marie et Pierre, les fondateurs de PermaBio et les invite à la revue de fin de sprint. Il est essentiel qu'ils soient là, car la revue est un moment d'inspection collectif de l'état du produit. Il en profite pour leur parler du paiement des plats, l'objectif du prochain sprint.

2.4.2. Le cycle symbolique du don basé sur la réciprocité

Pierre, qui a du pouvoir et de la légitimité, devient un interlocuteur préférentiel de Lucas pour les prises de décision.

Dans ce schéma, l'idée est que le PO s'appuie sur lui pour définir les priorités sur les gros morceaux. Ils pratiquent ensemble ce qu'on peut appeler un affinage à haut niveau. Ils s'accordent sur la stratégie.

Du point de vue du paradigme du don (DDRR) :

- Lucas demande de l'aide.
- Pierre qui lui donne de son temps et de ses connaissances.
- Lucas reçoit le conseil.
- Lucas rend en faisant un produit qui marche, ce qui satisfait Pierre qui aura envie d'aider à nouveau.



On décèle dans cette relation un aspect fondamental du don, la **réciprocité**.

2.4.3. Les cycles diaboliques avec les parties prenantes

Dans les relations avec les parties prenantes, les situations menant à des cycles diaboliques (IPRG) sont nombreuses et fréquentes.

Voici quelques cas qui auraient pu se produire dans la relation de Lucas avec Pierre et Marie :

- Pierre qui ignore les demandes de Lucas qui a des questions
- Pierre (il est DG de PermaBio) qui prend le pouvoir pour imposer un choix : imaginons qu'il veuille qu'on collecte des données pour les revendre. Lucas peut essayer de dire NON, ce qui est en phase avec ses valeurs. Si Pierre insiste, on rentre dans le cycle diabolique.
- Marie qui refuse de venir à la revue de sprint
- Pierre qui garde le droit de donner du travail aux développeurs sans passer par Lucas

2.5. Le don pour ne pas ignorer l'impact sur le vivant

2.5.1. Lucas se préoccupe des écosystèmes naturels

Lucas est sensible à la réduction des déchets et à leur recyclage. C'est pourquoi il a proposé la story compostage et qu'il la pousse.

Il se préoccupe également du prélèvement sur les écosystèmes naturels lié au développement du service.

Il réfléchit à tout ça en se promenant fréquemment au bord du canal ; dans ces lieux où il y a des arbres et de l'eau, il entre en résonance avec le vivant (et la musique qu'il écoute).



Figure 4. Les bonnes vibrations de Lucas en promenade

2.5.2. Le cycle symbolique du don basé sur la gratitude

PermaBio se définit comme un service de fruits et légumes à domicile. Ces fruits et légumes viennent des cultures de PermaBio ou de ses partenaires.

Si on n'a pas lu le dernier livre de Baptiste Morizot^[7], on va dire que PermaBio est producteur. Cependant, Morizot nous entraîne à bien saisir que c'est le vivant qui produit tout ça. Les maraichers ne produisent pas de tomates ou de chou-raves, ils aident, du mieux qu'ils peuvent, le vivant (la terre avec ses vers de terre et tous les autres organismes) à les produire.

C'est le don que nous fait le vivant.

Revenons à un cycle du don symbolique DRRR :

- PermaBio (dont Lucas est le PO) plante des graines, ce qui correspond à la demande
- Le vivant produit des fruits et légumes (il donne)
- PermaBio récolte (reçoit)
- PermaBio rend à la terre le compost provenant des fruits et légumes.



Comme cela se pratiquait dans les sociétés primitives, PermaBio fait preuve de **gratitude** en rendant au vivant, afin que celui-ci puisse continuer à donner.

2.5.3. Des cycles diaboliques avec le vivant

Les exemples d'ingratitude envers le vivant ne manquent pas : cela se manifeste par de multiples déviances qui conduisent au cycle IPRG.

En voici quelques-unes :

- ignorer le besoin de diversité du vivant en lui injectant des pesticides ou en mettant des graines OGM
- prendre tout ce qui est produit pour l'envoyer à l'autre bout du monde
- refuser les fruits et légumes mal calibrés

- garder les surplus plutôt que de les distribuer à ceux qui en ont besoin.

Si on reste sur le développement du service, des cycles diaboliques proviennent des technologies utilisées qui bien souvent dégradent les écosystèmes vivants.

Dans le cycle diabolique du don, c'est Prendre qui reflète les prélèvements de l'industrie du numérique :

- l'extraction des métaux rares pour les composants utilisés dans les matériels,
- la consommation d'énergie pour développer et exploiter les services.

Certes la technologie peut aussi avoir un impact positif et il est envisageable de mieux équilibrer la relation, c'est l'ambition de l'économie symbiotique^[8].



Cependant, on constate beaucoup d'**ignorance** dans les relations des entreprises du numérique envers le vivant, ignorance parfois camouflée avec du green-washing marketing.

2.6. Le don au sein de l'équipe agile

Une équipe agile est constituée de 5 à 7 personnes, avec deux rôles significatifs, celui joué par Lucas et celui de Scrum Master, tenu par Sarah. Elle n'est vraiment agile que grâce à la richesse des relations entre ses membres. Voici deux exemples, toujours avec le point de vue du Product Owner.

2.6.1. Lucas coopère avec Sarah pour l'affinage

Comme tous les jeudis, Lucas rencontre Sarah pour préparer l'affinage du backlog que Lucas anime ensuite avec l'équipe. Ils conversent ensemble afin de s'assurer qu'il y aura de quoi alimenter l'équipe pour le prochain sprint, pour éviter les à-coups dans le rythme.



Figure 5. Sarah et Lucas

Lucas annonce à Sarah qu'il va présenter la nouvelle story sur le paiement. Sarah lui indique qu'elle ne va pas être facile à réaliser dans le sprint car Mehdi qui connaît le mieux le sujet est pris par une formation.

2.6.2. Lucas fait la paire avec Nathan

Pendant le sprint, Lucas rencontre souvent le reste de l'équipe. Il participe à la mêlée et c'est une bonne habitude qu'il reste discuter avec les développeurs à la suite de ce rite quotidien. Ces conversations permettent de confirmer que la fonction à développer a été bien comprise par tous.



Figure 6. Lucas déroule une story sur le poste de Nathan

Il lui arrivera de constituer un binôme avec un développeur comme Nathan, pour que celui-ci lui montre, sur son environnement de développement, le fonctionnement de la story sur laquelle il travaille et lui fasse part d'une nouvelle idée pour l'améliorer.

2.6.3. D'autres points de vue sont nécessaires

Ces deux exemples pris dans une journée de Lucas nous suggèrent que les dons fréquents et sincères au sein de l'équipe sont efficaces.

Cependant, pour vérifier que le paradigme du don s'étend à l'équipe agile, nous allons examiner d'autres points de vue :

- celui du Scrum Master qui va nous situer le don dans son initiation au **coaching** ainsi que dans deux rites, la **mêlée** et la **rétrospective**),
- celui d'une équipe qui pratique le **mob programming**, la quintessence du don en équipe.

C'est l'objet des chapitres suivants.

[1] On peut la visionner en [VF sur Youtube](#) ; Henrik qui ne lésine pas sur le don a également publié une animation pour expliquer [le réchauffement climatique](#).

[2] Voir <http://agileradical.org/>

[3] L'anthropologue Marcel Mauss a écrit cet essai en 1925

[4] Publié en septembre 2019, Actes Sud

[5] selon les zones inspirées de la permaculture, qui sont présentées page 60 dans Scrum, pour une pratique vivante de l'agilité, 2018, Dunod

[6] Publié en 2019, Dunod

[7] Raviver les braises du vivant, 2020, Actes Sud

[8] [L'économie symbiotique](#), Isabelle Delannoy, 2017

Chapitre 3. Le don du point de vue du Scrum Master

Avant de devenir Scrum Master dans sa nouvelle entreprise, Sarah était chef de projet, manager de plus de 30 personnes.

3.1. L'adonnement improductif de Sarah

Dans son ancien poste, elle avait su créer une bonne ambiance dans le service par la mise en place de pratiques collaboratives tirées de Scrum, mais se heurtait à une certaine inertie, un manque de relais. Elle se donnait pourtant à fond, mais se sentant un peu seule et limitée dans ses initiatives, elle avait ressenti le besoin d'évoluer vers un contexte où elle pourrait avoir plus d'entrain.

L'irruption de la période du confinement, lui laissant un peu de temps d'introspection — gagné sur les déplacements — a permis de préciser ses envies :

- elle sait qu'elle veut continuer à aider des équipes ou groupes de personnes à surmonter leurs difficultés
- elle a besoin de se sentir plus utile ailleurs, autrement, avec créativité
- les choses évoluent quand-même dans le bon sens dans son équipe et il est peut être temps...

Du point de vue du don, nous sommes dans un cas — plutôt fréquent — d'adonnement d'une personne sans que le cycle se poursuive. Ce que donne Sarah n'est pas reçu et encore moins rendu.

3.2. Sarah est accompagnée d'un coach agile

Quelques semaines plus tard...

Sarah est désormais Scrum Master dans sa nouvelle entreprise.

Le coach c'est Jérémy. Ce que lui donne Jérémy à son arrivée, ce n'est pas l'explication de texte de la fiche de poste de Scrum Master, ni la liste des contraintes et difficultés auxquelles elle va être confrontée, ni encore moins les profils psychologiques des personnes de l'équipe. C'est une conversation rassurante sur la façon dont pourrait se dérouler son intégration dans l'équipe, au delà de suivre le plan de formations et d'accepter tous les rendez-vous jusqu'à saturer son agenda ; Ils concluent leur premier entretien par la promesse d'une future conversation pour partager régulièrement leurs observations, et continuer à apprendre ensemble.

Ainsi, Sarah pourra se faire une idée personnelle de ce qu'elle peut apporter à l'organisation ; elle pourra gagner en confiance par des feedbacks fréquents avec Jérémy.

3.3. L'observation des relations dans l'équipe

Sarah a assisté, en tant qu'observatrice, à une réunion — très technique — de tous les développeurs au sujet de la nouvelle architecture à mettre en place pour assurer la transition vers le Cloud.

Pour mieux comprendre les interactions entre les personnes, la qualité de la communication — et donc de circulation des informations — dans l'entreprise, Sarah a tenté de relever les dépendances portant sur les savoirs et savoir-faire ainsi que les signes de freins à la fluidité des échanges. Elle s'est concentrée sur la nature des questions et des réponses pendant la réunion.

Le lendemain, au rendez-vous hebdomadaire, elle retrouve Jérémy pour lui faire part de ses remarques, synthétisées en 4 relations simples :

Qui demande quoi à qui

Les développeurs avaient tendance à se tourner vers Marco (le leader technique) pour acquérir des certitudes ou se rassurer sur la faisabilité technique. Ensuite, les sujets de discussion revenaient souvent aux prises de décision par des personnes extérieures à la réunion (Product Owner, directeur). C'est d'ailleurs pour chercher le consensus qu'ils en étaient venus à tous se rassembler mais aussi par peur de rater quelque chose d'important.

Qui donne quoi à qui

Marco avait alterné entre ignorer la question en renvoyant aux pages de l'étude qu'il avait rédigée et refuser de répondre sur certaines pistes de recherche en cours ou non explorées, voire donner des informations directives dans la mise en oeuvre de changements au cas par cas avec un effort plus ou moins substantiel demandé à chacun.

Qui reçoit quoi de qui

Les développeurs avaient été très réceptifs aux informations et avis partagés, utiles pour évaluer risque, effort et urgence à organiser leur dépendance vis à vis de nouveaux éléments, composants et compétences techniques.

Qui rend quoi à qui

Les personnes qui avaient participé au développement de premiers composants se sont engagées à les publier avec une qualité et une documentation suffisante pour que les équipes produits puissent évaluer la difficulté d'intégration et rendre à leur tour un feedback plus objectif et constructif. Pour rassurer les développeurs, Marco s'est proposé pour donner une formation synthétique sur l'essentiel que chacun doit savoir pour mieux contribuer aux échanges.

Cette base d'observations factuelles a permis à Sarah de se prémunir du jugement des personnes et l'a préservée des biais, comme fabriquer automatiquement des problèmes et des solutions pour les résoudre.

La description du comportement des interactions dans le langage du don, tout en étant facile à faire par Sarah, s'est avérée puissante pour partager avis, ressentis et hypothèses sur les points forts et

les limites du système observé.

3.4. La demande dans les relations de Sarah avec l'équipe

En tant que Scrum Master, Sarah est là pour encourager l'équipe à suivre le cadre Scrum. Dans le détail, la façon pour elle d'accompagner l'équipe, d'y trouver sa place, restait à clarifier. Pour cela, il était important de bien connaître la demande de l'équipe.

Sarah a procédé par une question explicite adressée à tous les membres de l'équipe :

Qu'attendez-vous précisément du rôle de Scrum Master, maintenant et dans les semaines qui viennent ?

De cette façon, Sarah a reçu des réponses explicites sur :

- ce qui était vraiment important que l'équipe réussisse à faire sans son aide (en auto-organisation),
- ce en quoi elle — en tant que Scrum Master — pouvait aider.

En réponse aux difficultés révélées par la dernière réunion, tous se mirent d'accord sur l'intérêt de donner de la clarté et de la visibilité sur ce chantier technique, en le découpant afin de rendre sa compréhension collective et sa réalisation plus rassurante et efficace. Ils insistèrent sur le besoin de formation et demandèrent à Sarah de les aider en ce sens.

En retour, elle a répondu en s'engageant sur 3 actions :

- Avec Marco, je travaillerai sur la présentation de l'évolution de l'architecture technique avec la clarification des enjeux pour la prochaine démo de sprint.
- À mi-sprint je ferai un point avec Lucas (le PO) pour finaliser le scénario de démo et aussi préparer la réunion d'affinage.
- La veille de la démo, je viendrai vous voir chacun une heure pour préparer le scénario que nous pourrons présenter ensemble.

3.5. Le rite de la mêlée quotidienne pour raviver la dynamique du don

Comment passer de "Refuser de demander ou refuser de recevoir de l'aide" à "Demander et accepter l'aide" ?

Un comportement que Sarah avait pu observer dans l'équipe était qu'une personne refusait de se plonger dans le problème du voisin, tant qu'elle n'avait pas elle-même fini sa story, son travail de plusieurs jours. Lorsque deux personnes étaient bloquées, cela pouvait prendre plus d'une journée avant qu'elles pensent, ou soient conduites à penser, qu'il serait peut être mieux de s'y mettre à deux.

À la mêlée quotidienne, Sarah s'est proposée pour travailler en binôme, car même si elle n'y

connaissait encore pas grand chose, c'était une bonne occasion pour apprendre.

Un Scrum Master, plus encore qu'un nouveau venu, tire parti de son incompétence technique (réelle ou simulée) pour faciliter l'apprentissage collectif.

C'est peut être ici qu'est née la rumeur du Scrum Master bi-dons : il ne sait pas grand chose a priori et pourtant la vie est plus facile quand il est là.

Au fur et à mesure, d'autres comportements ont émergé. Après la mêlée quotidienne, quelques rendez-vous étaient pris dans la matinée, pour partager un point de vue de conception, avoir une meilleure compréhension d'une story avec le PO, aider le préposé au support à reproduire un bug, ou bien, avec Marie du marketing, co-créeer une vidéo pour faire la promotion de la toute dernière fonctionnalité du produit.

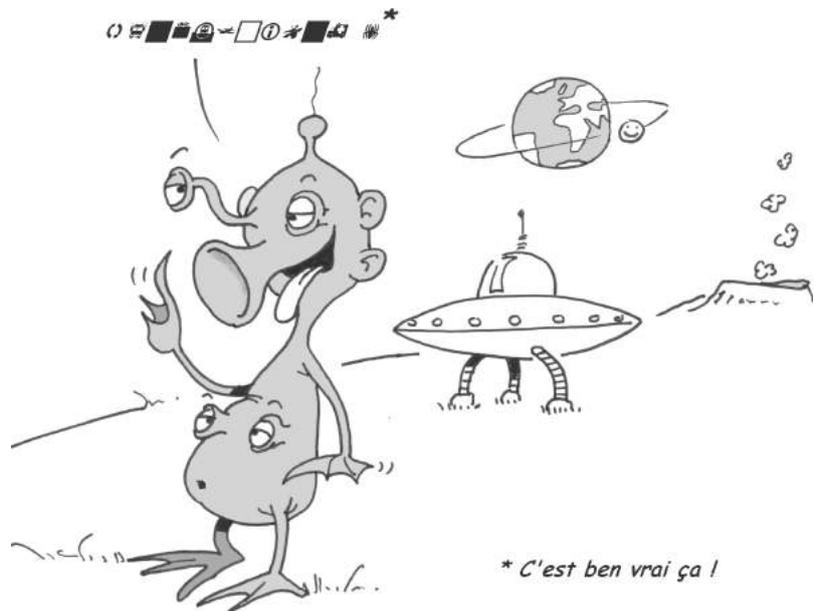


Figure 7. Quatre yeux valent mieux qu'un ;-)

C'était le signe qu'il devenait de plus en plus facile de demander, donner, recevoir et rendre au quotidien, au coeur de l'équipe. Au fur et à mesure que la connaissance individuelle des sujets se transformait en un savoir commun, les sentiments d'urgence, de perte de temps et de crainte lors des livraisons s'estompaient, l'équipe gagnait en confiance.

3.6. Le rite de la rétrospective pour rallumer le feu du don

La rétrospective est peut être un des rites Scrum dont la puissance symbolique est la moins comprise. C'est d'ailleurs à la façon dont l'équipe effectue (ou pas) les rétrospectives que l'on peut se faire une idée assez précise du sens qu'elle donne au mot "équipe". Plutôt que de livrer à Sarah la recette de la rétrospective magique, ou une nième nouvelle recette, Jérémy a pris le temps de lui montrer le sens de quelques ingrédients.

Encore une rétro ? Pourquoi faire ?

L'initiative de faire une rétrospective vient généralement du Scrum Master, et à travers lui, de

l'intention de faire du Scrum pour en tirer des bénéfices. Pour faire court, la rétrospective est un temps limité que l'équipe se donne à la fin de chaque sprint, pour réfléchir à comment s'améliorer. Si le retour sur investissement de ce temps n'est pas bon, ou n'est pas perçu comme suffisant, l'intérêt faiblira, inévitablement. D'où l'importance de donner du sens à ce rite en permettant à chacun d'y contribuer positivement.

3.7. Un cadre propice au don

Pour animer sa première rétrospective avec l'équipe, Sarah a jugé utile de démarrer par rappeler la "Prime Directive" écrite sur le tableau papier :

Indépendamment de ce que nous découvrons, nous comprenons et nous croyons sincèrement que chacun a fait du mieux qu'il pouvait, compte tenu de ce qu'il savait à l'époque, de ses compétences et de ses prérogatives, des ressources disponibles et de la situation du moment.

— Norman L. Kerth, Prime Directive

La "Prime Directive" pose le cadre idéal pour que l'équipe réalise une "bonne rétrospective".

Elle peut se reformuler dans un langage de dons :

- Nous avons donné le meilleur de nous même, compte-tenu de ce qui nous a été donné de savoir ou pouvoir faire.
- Et quand bien même nous pourrions avoir des doutes à ce sujet, nous acceptons et recevons ce résultat pour mieux rendre compte de nos observations et découvertes.

Cette directive, signe ainsi la prescription au sens juridique du terme, en déclarant inutile et vaine la recherche d'éventuels fautifs ; Ce que certaines équipes traduisent dans leurs accords de travail par "No finger pointing".

La rétrospective est un pardon collectif, utile pour tourner la page et engager le collectif vers sa réussite.

3.8. La célébration dans le cycle du don

Il est important de célébrer les réussites, les résultats obtenus et aussi les apprentissages, pour transcender ces connaissances individuelles en savoir commun. Medhi, un développeur de l'équipe, a eu la chance de participer à l'Agile Tour Toulouse. Il fait part à Sarah, de ce qu'il a retenu d'utile à propos de la rétro. Fabien Pelous, qui nous a fait l'honneur de sa présence en keynote de l'Agile Toulouse, nous a éveillé sur tous les intérêts de la troisième mi-temps, au rugby.

La troisième mi-temps a lieu que l'on gagne ou que l'on perde. Elle est presque plus importante dans ce dernier cas. C'est le moment où l'on peut se parler, se dire les choses, de la passe réussie, ou du coup de pied raté, dommage. Elle aide à tourner la page, pour que le lundi matin nous ayons de nouveau envie de revenir à l'entraînement.



Figure 8. Un temps pour la rétro

3.9. Le pardon mais pas l'oubli

Pour Sarah et son équipe, ce n'est qu'après la rétrospective que l'on clôt le sprint, pour en commencer un autre. Mais avant cela, que faut-il retenir de celui qui se termine ?

Dans certains endroits, reconnaître la réussite ne se pratique pas si souvent et se relativise facilement à la baisse :

OK, on a fait le job. Et encore le résultat est loin d'être parfait.

Sarah sait combien ce sentiment d'insatisfaction personnelle est nuisible pour elle, et donc pour les autres. Curieusement, elle sait davantage comment valoriser une démo, révéler ce qui s'est passé de positif dans l'équipe. En renforçant l'équipe de cette reconnaissance réciproque, elle aide l'équipe à tirer des enseignements de ses erreurs.

La rétro ce n'est pas "Thanksgiving" : une histoire de dindons plus que de don véritable

Ce qu'il restera comme trace visible de la rétrospective, et peut-être du sprint tout entier, c'est la photo de la "véloce embarcation" et de ses ancres, prise pour métaphore de la performance de l'équipe, pour être rangée dans le dossier partagé. Mais l'atelier ludique et créatif aura surtout permis de passer un moment convivial et utile pour réellement apprendre sur soi, sur les autres.

Ce petit temps entre parenthèses, entre passé et futur, est en fin de compte celui où l'équipe écrit son histoire et définit son rêve. Elle se donne du sens, de la confiance, du courage, nécessaire aux changements qu'elle peut opérer sur elle-même et qu'elle voudrait donner à voir dans le monde.

Un sprint est comme un calendrier de l'avent : un voyage à compte à rebours des jours restant à parcourir avant la prochaine épiphanie.

Chapitre 4. Mob Programming & Don

Le Mob Programming nous a été donné par Woody Zuill (avec la contribution de Llewellyn Falco).

Woody définit ainsi le mob programming:

Toutes ces brillantes personnes travaillant sur la même chose, au même moment, dans le même espace et sur le même ordinateur.

— Woody Zuill, Définition Mob Programming

4.1. Le mob pour apprendre

Le mob a d'abord été expérimenté pour apprendre, en tant que cadre favorisant la transmission des connaissances au sein d'un groupe de programmeurs et programmeuses, et plus largement toute autre personne.

C'est un superbe véhicule pour l'apprentissage dans lequel les personnes échangent : leurs savoirs, leurs points de vue, les différentes approches qu'ils peuvent avoir à la résolution d'une problématique. Les débutant-es peuvent également apporter un regard neuf et décapant, le plus souvent, de simplicité éliminant ainsi cette complexité accidentelle au sein des systèmes en développement.



Le mob programming requiert un certain apprentissage ...

Figure 9. Illustration de Patrice Courtiade

Nous aimons rappeler cette citation anonyme :

Ce qu'il y a de bien avec la connaissance, c'est qu'une fois donnée, on ne peut pas la reprendre !

— Anonyme

4.2. Le mob pour développer en équipe

Après avoir démontré ses apports de générosité dans le domaine de l'apprentissage, Woody et son équipe ont essayé cette approche pour développer les fonctionnalités d'un logiciel.

Ils l'ont mis en oeuvre (ou pour le moins testé de façon informelle) pour tenter d'amener de

nouvelles fonctionnalités sur une base de code avec laquelle l'équipe se débattait.

Cette approche ayant apporté des progrès significatifs pour la reprise du contrôle de ce code, le Mob Programming était né.

Le mob donne à l'équipe une force collective et un courage sans équivalent lui permettant d'affronter de grands défis, ceux qui prennent parfois la forme de monstres émanant des sombres contrées du *Code Legacy*.

C'est grâce à la richesse des idées et des différents savoirs que de nouvelles voies sont ouvertes, considérées, implémentées et évaluées.

4.3. Les rôles du mob et le cycle du don

Nous distinguons deux rôles au sein du mob : le driver et le navigateur.

Le **driver** est le développeur ayant le clavier, il est responsable de l'écriture du code, il écoute et retranscrit les suggestions ou idées proposées en lignes de code.

Le **navigateur** pose le problème, écoute, fait émerger les idées, donne la direction du développement.

Dans un premier temps, les autres participants sont membres du mob et peuvent intervenir afin de questionner ou pour faire don de leurs idées. Dans la phase d'apprentissage, il leur est conseillé de lever la main pour demander la parole afin s'assurer que les échanges soit toujours un dialogue avec le navigateur. C'est la meilleure façon de réaliser un don ou de poser une question, l'ensemble des participants (driver, navigateur et autres membres) peuvent ainsi tous recevoir la proposition ou le questionnement. Lorsque le mob gagne en maturité, les membres deviennent tous des navigateurs à tour de rôle.

Examinons le mob au regard du cycle du don :

DEMANDER - DONNER - RECEVOIR - RENDRE

Le navigateur peut demander, à tout moment, si quelqu'un a une idée pour résoudre le problème, il donne alors la parole et l'écoute qu'il convient au donateur, recevant l'idée il va la rendre intelligible pour tous et notamment auprès du driver.

Une séance de mob réussie déroulera de multiples cycles symboliques de dons.

Nous sommes alors à l'opposé du cycle diabolique

IGNORER - PRENDRE - REFUSER - GARDER

parfois constaté entre des développeurs qui :

- ignorent les différences de niveau et de compétences,
- ne prennent en compte que leurs habitudes de codage,

- refusent d'expliquer leurs choix de design,
- afin de garder une connaissance exclusive de ce qu'ils ont développé.

4.4. Les valeurs du mob

Quand Woody et son équipe ont décidé de poursuivre l'aventure du Mob Programming, ils ont estimé qu'il était essentiel de définir un cadre afin que cette approche du développement du logiciel puisse pleinement et durablement s'épanouir ; cela commence avec des valeurs et un accord.

Le mob tient dans ces trois valeurs :

- Bienveillance,
- Respect,
- Considération.

Et pour que chacun puisse trouver sa place dans le mob, il convient de respecter cet accord :

Traitez-vous les uns les autres avec bienveillance, considération et respect.

— Woody Zuill, Valeurs du Mob



Figure 10. Un Accord

Cet accord donne à chacun sa place pour s'exprimer.

Il est dit qu'être un bon navigateur c'est, entre autres, amplifier les idées murmurées.

Nous établissons ainsi, un cadre sécurisant pour tous, cela permet à chacun de demander une explication ou de l'aide afin de mieux comprendre quel est l'intention de ce que nous réalisons ou des éclaircissement sur telle ou telle partie de code. Ce cadre permet de donner des idées : de

solution, de design...

Le driver peut également donner quelques astuces de raccourcis clavier ou plus simplement des astuces du langage de développement utilisé. En faisant ainsi don aux uns et aux autres de son savoir, c'est le cycle du don qui est enclenché, les donateurs se sentent alors redevables de partager leur savoir.

DEMANDER - DONNER - RECEVOIR - RENDRE

est l'énergie qui anime le mob programming.

4.5. Le mob pour amplifier la capacité d'écoute

Le Mob Programming nous donne à travailler notre capacité d'écoute, à la développer afin d'être attentif aux uns et aux autres pour que chacun puisse amener toute suggestion. C'est par cette capacité d'écoute que chacun reste attentif : aux demandes d'aide, aux idées des donateurs, aux dons reçus.

Je me rappelle qu'au cours d'une de nos sessions, une proposition d'amélioration de notre code a émergé, mais en tant que navigateur je n'ai pas réussi à saisir l'idée. Constatant que notre ami développeur avait quelques difficultés à formuler ce qu'il percevait, j'ai demandé si un autre membre du groupe pouvait reformuler la proposition.

Personne n'ayant réellement compris l'idée, il eut été facile de poursuivre le chemin que j'avais à l'esprit ou pire de proposer au développeur de prendre le clavier (de devenir driver) et ainsi d'écrire **sa solution**.

Au lieu de cela, nous sommes restés attachés à appliquer ce principe fondamental du [strong-style pairing](#).

*Pour qu'une idée passe de votre tête à l'ordinateur, elle **DOIT** passer par les mains de quelqu'un d'autre*

— Llewellyn Falco, Strong Style Pairing

Pour être respectueux de ce principe, le développeur ayant eu l'idée d'amélioration a donc pris le rôle de navigateur et nous lui avons demandé d'être le plus expressif possible. Lorsque le driver est perdu — et plus encore si le reste du groupe aussi — alors le navigateur se doit de réduire le niveau d'abstraction de son langage pour en venir à décrire :

- l'intention,
- puis la localisation
- et enfin les détails d'implémentation permettant à chacun d'assimiler la solution.

Cela permet au driver d'écrire le code souhaité. Dans cette situation le navigateur en vient à énoncer le numéro de ligne et le code exact qu'il souhaite voir.

Mettant en oeuvre cette approche nous avons pu voir se concrétiser une excellente idée simplifiant

le code tout en permettant à l'ensemble du groupe d'en comprendre la portée.

Le **Strong Style Pair Programming** énoncé par Lewellyn Falco offre une porte vers le monde de la compréhension et du partage, un levier rendant intelligible les idées ou concepts perçus par un esprit.

DEMANDER - DONNER - RECEVOIR - RENDRE

sont des signes d'un mob bien vivant et sein, bref d'un mob en pleine santé.

4.6. Le mob et les choix de conception

Et si une pluralité d'idées émerge ? Eh bien il convient de considérer chacune.

Woody l'exprime ainsi :

Avec le Mob Programming, nous ne gaspillons pas beaucoup de temps à décider laquelle des deux (ou plus) idées nous allons essayer - nous faisons simplement l'une ou l'autre, nous évaluons le résultat, puis nous considérons si nous devons essayer l'autre. Cela permet aux idées de se transformer, de fusionner, d'évoluer et d'émerger.

— Woody Zuill, Respecter les idées, les évaluer

Appliqué au sein de votre session de Mob Programming (ou de Pair Programming), ce principe évite bien des frustrations.

Vous serez souvent étonné de constater que le groupe s'arrêtera à la première idée implémentée, car au final il y avait peu de différences entre les deux idées (ou plus) présentées.

Il est difficile d'échanger au niveau des concepts et des idées portant sur des formes d'implémentation alors qu'il est beaucoup plus aisé d'en discuter sur un code existant concrétisant l'expression des idées.

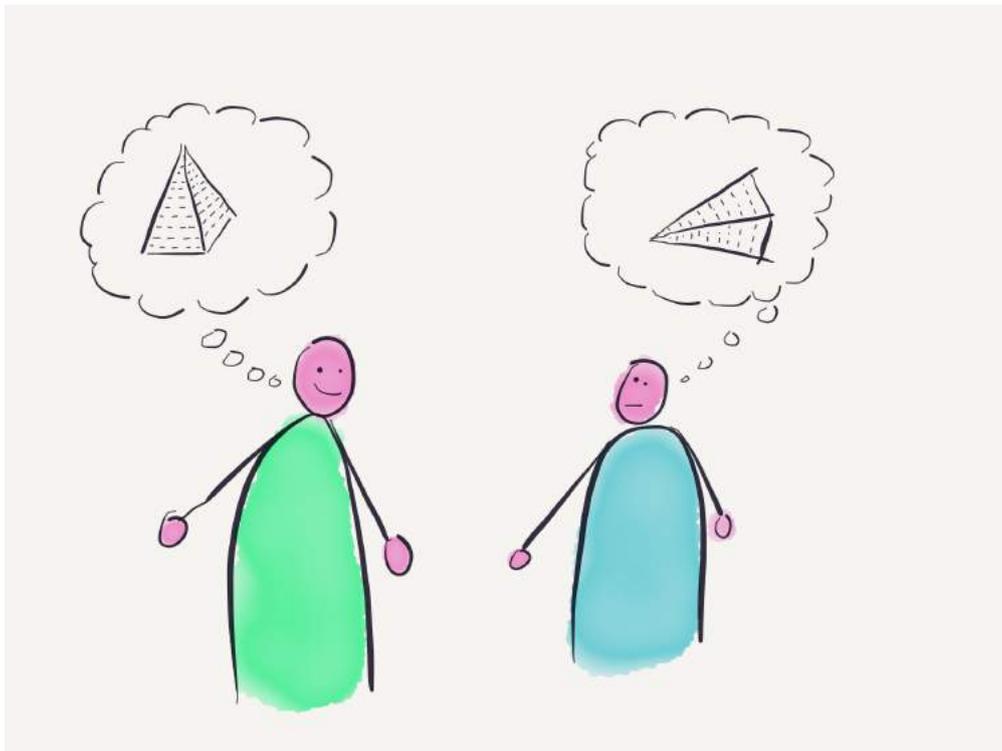


Figure 11. Implémenter et Discuter

J'ai régulièrement pu constater que deux idées se rejoignent ou finissent par fusionner.

Bref nos idées sont vivantes, évolutives et se transforment par les apports des uns et des autres. Le temps du mob est un temps de construction d'un bien en commun, la base de code réalisée est alors comprise par chacun et d'une grande qualité.

Le temps du mob est un temps pour DEMANDER de l'aide, pour se DEMANDER si nous ne pourrions pas explorer telle idée. C'est surtout le moment pour la partager et en faire DON, l'équipe la RECOIT alors avec bienveillance, l'implémente et en vient souvent à la RENDRE meilleure.

4.7. Mob programming et rétrospectives

Nous apprécions grandement les rétrospectives, c'est un moment privilégié pour demander, signifier, remercier et donc réaliser des dons et contre-dons. Le mob programming n'échappe pas à cette règle, il fonctionne d'autant mieux lorsque des rétrospectives sont réalisées régulièrement après chaque séance.

Mark Pearl dans son livre *Code with the Wisdom of the Crowd* (édition [pragprog](#)) conseille une rétrospective d'environ quinze minutes après une heure trois quarts de mob programming. Cela permet de faire tenir dans une time box de deux heures, une séance de mob programming et une rétrospective, en faisant attention à ne pas oublier les pauses.

Dans cette courte rétrospective nous cherchons, comme pour toute rétrospective, à mettre en exergue les faits dans un premier temps, ensuite les aspects positifs puis les points négatifs afin de déterminer ce que nous souhaitons améliorer sur la prochaine time box.

Par bien des facettes le mob programming est un véhicule facilitant les membres d'une équipe à entrer dans le cycle symbolique du don.

Chapitre 5. Conclusion

- Nous avons eu l'idée de cet article sur le don suite à la DEMANDE de Robin pour une contribution à un calendrier de l'avent agile.
- Nous vous le DONNONS avec plaisir, avec cette publication sur la case 1, en introduction aux 24 autres dons.
- Si vous êtes arrivé-e jusqu'ici (merci pour votre lecture, nous avons conscience que c'est un don bien long, mais nous avons donné le meilleur de nous-mêmes), c'est que vous l'avez REÇU !
- En tant que donataire, ce que vous pouvez RENDRE (et ainsi devenir donateur à votre tour), c'est à vous de le décider.

Voici quelques suggestions sur quoi faire avec cet article :

- nous donner du feedback, pour peut-être enclencher un nouveau cycle pour un article en V2 qui serait plus facile à lire pour les lecteurs et les lectrices,
- expérimenter le paradigme du don étendu dans votre équipe (voire dans votre vie) comme moyen de lutte contre le faux-agile,
- en faire don à d'autres personnes.

Et bien sûr, revenir sur agile christmas tous les jours jusqu'à Noël pour y découvrir de nouveaux dons.