Peut-on parler de DevOps sans parler d'Agilité

Je m'appelle Vincent et je ne suis pas DevOps. Alors je sais ce que vous vous dites : "M'enfin, cet homme n'est pas Devops, d'où est-ce qu'il vient parler d'un sujet qu'il ne connait pas, tu vas voir quel ROTI je vais lui mettre à celui-là :"

Oui, je ne suis pas DevOps... Je suis Ingénieur Système Agile... ou ISA, le fameux mouton à 5 pattes que s'arrachent les ESN, puisque l'ISA bêle...

Bref, je ne suis pas DevOps, tout simplement parce que DevOps n'est ni un rôle, ni un métier, contrairement à la pensée populaire qui voudrait faire d'un "DevOps" le facilitateur entre l'équipe Dev et l'équipe Ops. Si vous pensez ça, c'est que vous n'avez pas compris ce qu'était DevOps... Mais ce n'est pas grave, laissez-moi vous expliquer ce qu'est DevOps en vous comptant son histoire...

Il était une fois DevOps

Il était une fois, dans un royaume pas si éloigné du notre, vivait Patrick Debois. Patrick était ce qu'on appelle un OPS. Dans ce royaume, son rôle à lui était de déployer et maintenir les applications developées par les Devs. Mais ce n'était pas chose évidente, car si les Devs était devenus agiles, personne ne prenaient en compte la maintenance en condition opérationnelles. Après-tout ce n'était pas le rôle des Devs de penser à ça, pensaient-ils. Patrick était perçu comme un Bastard Operator From Hell !

Pourtant Patrick voyait en l'Agilité la réponse à nombreux de ses problèmes et se prenait à rêver d'apporter plus d'Agilité dans la gestion des instrasutructures et des opérations.

C'est donc dans ce sens qu'il participa en 2008 à l'Agile Conférence.

Et comme il adorait l'Agilité, il y fut également attiré par une conférence du développeur Andrew Shafer. Malheureusement, Andrew, persuadé que le sujet n'interesserait personne, ne vint pas à sa propre conférence, laissant Patrick tout seul dans la salle. Heureusement, malgré tout, ils purent se rencontrer, échanger sur leur vision commune et fonder... L'Agile Administration System Guild... euh group

Une année plustard se déroule le Velocity O'Reilly, une conférence à laquelle s'intéresse particulièrement Patrick car un retour d'expérience présenté par Paul Hammond et John Allspaw de Flickr doit y être présenté sur comment les gentils développeurs et les gentils opérationnels y travaillaient main dans la main au quotidien et leur permettait de faire plus de 10 déploiements par jours avec une équipe de seulement 18 personnes. Mais malheureusement Patrick ne peut s'y rendre et est tout chagrin sur Twitter.

Suivant les conseil de Paul Nasrat, il organisa alors son propre événement : Un événement commun pour les developpeurs et les opérationnels... Les

DevOpsDays (Journées des Devs et des Ops). Cet événement fut un grand succès et les discussions continuèrent sur twitter avec le hashtag... #DevOps. DevOps était né...

Les 4 piliers DevOps

DevOps est donc un mouvement culturel issu d'Agile. Il vise à intégrer la partie opérationnelle dans la démarche Agile, ce qui fait particulièrement sens, car un logiciel fonctionnel, et un logiciel supervisé et maintenable.

DevOps s'articule autour de 4 ou 5 piliers. CALMS !

Culture : Dave et Hops sont dans un bobsleigh...

Dave et son ami Hopper, dit Hops, sont champions de bobsleigh. L'un sans l'autre, ils ne pourraient même pas atteindre leur objectif qu'est le franchissement de la ligne d'arrivée. En effet, Dave est le pilote, c'est à lui d'anticiper les virages et de changer de trajectoire. Sans lui, Hops ferait une sortie de piste au premier virage. Hops, lui est le freineur. En effet, il ne faut pas que le bobsleigh aille trop vite, sinon, il risque de perdre sa stabilité et de sortir de piste dès le premier virage.

Cela a pris du temps, mais Dave et Hops ont pu apprendre à se connaître grâce à leurs nombreux échecs. Ils ont développé une culture commune et surtout une confiance qui leur permettent d'anticiper les actions de l'autre.

Automatisation : Cétautomatix, le roi de la forge

En tant qu'Ops, Cétautomatix fournit des environnements pour ses collègues développeurs. Afin de s'assurer de la qualité du livrable fourni par l'équipe, il est décidé que le livrable puisse être testé sur 4 environnements. L'environnement de développement accueillera toutes les modifications et devra pouvoir être reconstruit facilement en cas de problème lors de l'étape de développement. L'environnement de test permettra de faire tester l'application et réaliser des benchmarks. L'environnement de recettage permettra aux clients de tester la solution et de renvoyer du feedback à l'équipe de développement. Le dernier environnement n'est autre que la plateforme de production.

Il doit donc mettre en place et maintenir toute l'infrastructure et déployer l'application, faire tourner des tests, et être capable de revenir en arrière en cas de problème.

Dans ces conditions, il est important d'automatiser un maximum de tâches, afin de lui faire gagner du temps et éviter des erreurs humaines qui pourraient avoir un impact négatif sur le travail collectif.

Cette automatisation permet aujourd'hui de mettre en place de véritables usines logicielles, appelées forges, permettant de faire de l'intégration et de la livraison continues. Lorsqu'une mise à jour est poussée par un dev dans l'environnement de test de l'équipe DevOps de Cétautomatix, tout est testé (code, sécurité, régression, benchmark, ...), puis poussé dans l'environnement de recette du client.

Mesures : « Qui ne mesure, jamais ne dure. » (Proverbe gaulois français)

En automatisant l'administration de son système d'information pour en faciliter sa gestion, Cétautomatix a du le complexifier... beaucoup. Du coup, il est donc devenu nécessaire de gérer ce système de façon plus agile, en ayant une approche empirique. Et qui dit empirique, dit mesures. Puisqu'il s'agit d'un système d'information et non d'un système humain, les retours se font au travers d'indicateurs et de sondes. Cétautomatix va donc monitorer son système pour le faire évoluer, en utilisant des sondes tant au niveau de l'infrastructure, ce qui lui permettra d'avoir une gestion au plus juste de ses ressources (Lean), qu'au niveau applicatifs.

Ces métriques serviront également aux développeurs pour améliorer leurs logiciels, et même, de plus en plus au métier, puisqu'en rapprochant le résultat de plusieurs remontées, de nouvelles informations pertinentes peuvent être mises en valeur et produire de la valeur commerciale. C'est sur ce point que se base le mouvement Big Data.

Partage : La Culture est le partage de valeurs et de savoir-faire

« En sociologie, la culture est définie de façon plus étroite comme « ce qui est commun à un groupe d'individus » et comme « ce qui le soude », c'est-à-dire ce qui est appris, transmis, produit et créé »

Le Mouvement DevOps est comme on l'a dit un mouvement culturel, ce qui signifie que les valeurs qui soudent les équipes travaillant avec l'état d'esprit DevOps sont apprises et transmises pour ensemble produire et créer des solutions. Cela passe par un un échange de savoir qui a entre autre permis aux Ops de travailler avec des systèmes de gestion de l'infrastructure versionné dans des fichiers déclaratifs et variabilisés à l'image de méthodologies plutôt utilisées dans le cadre du développement (on parle d'Infrastructure as Code), ou bien d'utiliser des conteneurs pour livrer les composants d'infrastructure. Les équipes sont aussi plus transparentes et les métriques récoltés sont partagés globalement (métiers, développeurs, équipe de sécurité, etc...).

Le partage, c'est aussi le partage entre pairs, sous forme de retour d'expérience (produit, manière de travailler ou de contourner les obstacles, best practices). De nombreux événements permettent à tous les maillons de la chaîne de production

IT partageant la Culture DevOps, de se retrouver, et de continuellement échanger pour progresser collectivement.

Des outils pas si DevOps que ça

On m'objecte souvent que ce qui fait le DevOps, c'est l'utilisation d'outils dit "devops". En réponse à l'un de mes articles on m'a envoyé l'infographie suivante. Je vais donc être très clairs sur ce sujet. **CES OUTILS NE SONT PAS DEVOPS**. Ils peuvent être des outils d'intégration, des outils de développement, des outils de déploiement, des outils de communication, de conteneurisation ou d'automatisation mais l'utilisation de ces outils ne font pas forcément de vous des DevOps. Laissez-moi vous illustrer mon propos avec un cas d'école suivant :

Didier, Fadia et Joann... utilisateurs d'outils "devops"

Didier est développeur. Il appartient à une équipe qui travaille sur un produit P. Cette équipe à son "PO" dédié, son "Scrum Master" dédié, son backlog dédié. Le code est poussé sans un dépôt Git, puis Didier exécute une pipeline Jenkins pour builder son composant, son image Docker, qui sera ensuite déployée dans un cluster Kubernetes.

Fadia est intégratrice. C'est elle qui est chargée de créer la/les pipelines pour les équipes de développeurs en fonction de leurs besoins (Créer des buckets Couchbase, et ajouter les configuration dans la pipeline, permettre à la pipeline d'utiliser SonarQube pour les tests automatisés, etc). L'équipe de Fadia recevant des demandes de différentes équipes de développement gère son propre backlog selon les principes Kanban. Chaque personne prends des tâches qui sont demandées à l'équipe d'intégration. L'équipe de Fadia à son "PO" et "Scrum Master" dédié. L'équipe de Fadia fournit du service à l'équipe de Didier.

Joann est lui dans l'équipe de production. Il s'occupe de l'exploitation de des solutions développées par la société dans laquelle travaillent Didier, Fadia et Joann. Il s'occupe du bon fonctionnement des applications, voire de l'évolution des pipelines mise en place par l'équipe de Fadia. L'équipe de Joann, de par ses contraintes de production travaillent selon les principes Kanban avec leur backlog dédié. Un référent est associé à chaque équipe de développement pour assurer une meilleure gestion des incidents et des évolutions. L'équipe de Joann n'a pas de "PO", mais à un "Scrum Master" dédié.

On voit bien que les outils utilisés inclus dans l'infographie. Pour autant, on est très loin de DevOps. Toutes les équipes sont silotés, ne partagent par les objectifs, et pire non pas réellement de gouvernance commune.

Quels outils pour DevOps ?

Lors de son intervention à Agile en Seine 2018, Guillaume Lepetit, ancien DSI de TF1 qui a mis en place DevOps dans ce groupe média, avait eu cette phrase délicieuse : *"Ils (les collaborateurs) n'allaient pas pouvoir faire du DevOps sans avoir les bases de l'Agilité."*.

Scrum comme boite à outils DevOps

Contrairement à ce qui est répendu un peu partout, Scrum N'EST PAS une méthode, ni moins une méthodologie Agile.

Tout d'abord, ce n'est pas une méthode ou méthodologie, car c'est un cadre de processus léger, une sorte de boite à outils proposant des pratiques recommandées et n'imposant que peu de choses comme ses rôles, et ses cérémonies, qui permettent le mieux travailler ensemble.

D'autre part, Scrum n'est pas Agile, car ce cadre de processus est apparu plusieurs années auparavant et a été conçu pour le développement, la livraison et la maintenance de produits complexes, y compris donc les projets non-informatique. Il est donc possible d'utiliser Scrum dans le cadre du développement d'une fusée ou d'une campagne marketing. Le mot Agile N'EST PAS MENTIONNE dans le guide Scrum

Alors pourquoi proposer Scrum comme boite à outils DevOps, si Scrum n'est pas Agile ?

D'après Diana Larsen et James Shore, auteurs du Modèle d'Aisance Agile, la première zone à franchir pour qu'une organisation devienne Agile, est là zone dite de Concentration ("Focusing"). C'est dans cette zone que l'on va se concentrer sur la production de valeur métier. Elle s'appuie notamment sur l'apprentissage qu'une équipe peut tirer de cadre de processus comme Scrum.

Les spécificités de Scrum sont donc toutes indiquées pour qu'une équipe puisse commencer à mettre DevOps en place :

En Scrum, il est nécessaire pour une équipe de regrouper l'ensemble des compétences nécessaires au développement du produit. Ce qui veut dire pas seulement les devs, pas seulement les devs + les ops, mais aussi les personnes compétentes en monitoring, en sécurité, réseaux, voire en UX également. On parle de compétences et non de personnes, les membres de l'équipe Dev pouvant être multi compétents.

Il faut bien comprendre que équipe de développement ne veut pas dire "équipe de developpeurs". L'autre notion à retenir, c'est que c'est toujours l'équipe dans son ensemble, qui est responsable de l'incrément produit potentiellement livrable, peut importe les profils qui l'a compose. Les Devs, les Ops, les Secs sont autant

responsable de l'incrément produit. Si le succès de l'équipe est collectif, il en est également de ses échecs.

Il n'y a donc qu'un seul backlog pour l'ensemble des tâches, DEV, SEC, OPS, QA et l'équipe partage les même objectif et chacun de ses membres doit être aligné sur la même vision produit dont le PO à la charge.

Scrum s'appuie sur 3 piliers : Transparence, Inspection et Adaptation.

Ainsi l'une des cérémonies les plus importantes en Scrum est la Rétrospective. Elle permet à l'équipe, à chaque fin de sprint, de réfléchir à l'amélioration continue du travailler ensemble, en se basant sur les expériences passées.

L'équipe Scrum acquiert donc, sprint après sprint, une culture commune, de la transparence et du partage, basé sur les mesures du passé. Et puisqu'elle doit être en mesure de pouvoir livrer l'incrément du produit fini (répondant à tous les critères d'acceptances à chaque fin de sprint), l'automatisation, et la sécurité en sont donc des pré-requis.

Scrum permet de mettre en place facilement les 4 piliers DevOps, c'est pourquoi ce cadre de processus peut être considéré comme boite à outils DevOps.

En mettant en place Scrum, Didier, Fadia et Joann vont donc se retrouver dans la même équipe, partager les même objectifs, avec un backlog unique avec un seul PO et un seul Scrum Master unique. Ils travailleront enfin ensemble et non séparé et obtiendront la mise en place de DevOps.