Bonjour,

Comme les années précédentes depuis sa création, Ausy et mon client m'ont permis d'assister à l'Agile Tour de Bordeaux ce vendredi. Ainsi, comme les années précédentes, je vais partager avec vous mes notes sur le déroulement de cette édition.
Les précautions d'usage s'appliquent bien sûr à cet exemplaire : il ne s'agit que de mes notes et n'ont aucune valeur officielle vis-à-vis de l'évènement.
Ceci étant dit, je peux à présent me concentrer sur mon programme de la journée.

Nous avons été accueillis cette année par l'EPITECH (http://www.epitech.eu/) et force est de constater que les conditions d'accueil ont vraiment été hors du commun. Sans compter la mise à disposition de toute l'intendance nécessaire pour diffuser en live sur dailymotion l'ensemble des sessions déroulées dans l'amphithéâtre. Bref, voici une école offrant un cadre d'études des plus agréables.

Pour revenir au programme, la journée a commencé comme il se doit par un mot d'accueil de l'association Okiwi (http://okiwi.org/) qui organise l'événement, puis d'une présentation des sponsors et partenaires qui permettent le déroulement de l'évènement dans ces conditions de confort.

C'est ensuite que les choses sérieuses ont commencé par une keynote d'ouverture de David Evans (http://uk.linkedin.com/in/davidevansagilecoach) intitulée « **What testers and developers can learn from each other** ».
Il s'agit d'un orateur exceptionnel qui nous guide ici dans une réflexion sur la notion de test et sa prise en charge par l'équipe de développement.
Je ne remonterai ici que quelques points relevés, mais je vous invite à guetter la mise à disposition des vidéos pour vous faire une idée par vous-même.

- Ne pas avoir peur de se remettre en question, même et surtout lorsque l'on est expérimenté.
- Toujours essayer de nouvelles choses et convaincre son management de nous laisser essayer.
- Accepter et faire accepter de se planter de temps en temps et en profiter pour apprendre de ses erreurs.
- Il existe plusieurs stratégies de tests (manuels, semi-automatiques, automatiques).
- Il existe plusieurs techniques de tests (unitaires, d'intégration, fonctionnel, d'acceptation …).
- Les développeurs sont les premiers conscients que le fait que les tests soient OK n'est pas suffisant.
- Il n'y a pas de réponse simple à la question : « Qu'est-ce qu'un bon test ?». C'est une réflexion à avoir en fonction du contexte.
- Les tests ne garantissent pas la qualité. Ils ne font que surveiller et prévenir mais il est indispensable d'en assurer le suivi.
- Steeve McConnel « Penser que l'on peut améliorer la qualité en ajoutant des tests revient à penser que l'on peut perdre du poids en se pesant souvent ».
- Le test correspond à une mesure et un feedback. Il s'agit d'une activité sans fin.
  On n'a jamais fini de tester. On décide juste de s'arrêter à un stade que l'on juge acceptable en fonction des risques considérés.
- Kent Beck « Il faut considérer que ce qui n'est pas testé ne fonctionne pas ». ➔ Invitation au TDD.
- Il est nécessaire de choisir ce qui doit être testé et ce qui ne le sera pas, mais garder en tête que si une feature ne doit pas être testée (pour des questions de temps ou de moyens), elle ne devrait pas être développée.
- L'erreur est humaine, il s'en produira toujours. La recherche d'un fautif est une perte de temps et mène toujours à une impasse. Il faut se concentrer sur comment éviter qu'elle ne se reproduise.
- Un rapport de bug n'est qu'une information subjective tant qu'aucune action n'est prise pour y répondre.
- Un bug est la matérialisation d'un manque de test. Toute prise en compte d'un bug devrait commencer par la mise en place d'un test.
- Pour autant, il revient trop cher de *tout* tester. La valeur d'un test s'évalue surtout vis-à-vis du temps qu'il nous fait gagner sur la mise en œuvre d'action (temps de réponse à la détection d'une erreur).
- Les tests ne peuvent pas être uniquement pilotés par l'IHM.
- Le TDD ralentit le développement autant que des passagers ralentissent un bus. Pour autant la raison d'être d'un bus n'est pas d'aller vite mais bien de transporter des passagers.
- Nous avons plusieurs niveaux d'angoisse qui impliquent différents niveaux d'efforts sur les tests. Ces niveaux peuvent évoluer dans le temps « *Value can change* ».
- Il ne s'agit pas de juger la qualité des tests par le nombre de testeurs. Le test devrait être la préoccupation de tous. C'est une casquette que l'on devrait tous accepter par nature.

J'ai ensuite participé à un atelier de **management visuel** piloté par Xavier Quesada (http://www.xqa.com.ar/visualmanagement/). Une seconde session en anglais, mais le jeu en vaut la chandelle. Sans compter que je suis en cours d'élaboration du taskboard de la nouvelle équipe dans laquelle j'officie depuis peu.
Il s'agit donc d'un atelier qui tente de nous faire découvrir comment illustrer visuellement un projet selon 3 « méthodes » : XP, Scrum, puis Kanban.
C'est bien cette dernière qui m'intéressait le plus, mais nous ne sommes malheureusement pas parvenus à cette troisième phase dans les 2 heures imparties (léger problème de timeboxing).
Voici tout de même quelques éléments retenus des conseils de management visuel prodigués :

- Il faut visualiser ce qui doit être fait.
- Un support physique est plus simple et plus vite appréhendé qu'un support numérique. De plus, il apport un gain en visibilité de l'extérieur (les parties prenantes).
- Un taskboard doit avoir principalement 2 qualités :
    - L'équipe doit aimer l'utiliser
    - Il doit être particulièrement visuel sur les aspects importants
        - Reste à faire
        - En cours OK ou pas
- Les tâches restantes à accomplir sur un élément (une story par exemple) peuvent être matérialisées par des petits post-it (à la manière des impediments) plutôt que par la création de colonnes du taskboard. Surtout s'il ne s'agit pas d'étapes obligatoires pour l'ensemble des éléments du taskboard.
- Il peut être créé une zone de blocage en bas de chaque colonne.
- Matérialiser une colonne de tests implique souvent qu'ils ne sont pas considérés comme devant faire partie du développement, qu'ils ont une nature différente, ou qu'ils arrivent de façon trop tardive.
- Xavier préconise donc l'utilisation des 3 colonnes traditionnelles (TODO, IN PROGRESS, DONE) avec un ensemble de petits post-its qui matérialisent l'avancement.
- Le burndown sert à visualiser l'écart entre le déroulement de l'itération et les prévisions.
- Le taskboard peut être géré par un seul niveau de granularité (tel qu'en XP où les stories ont une petite granularité régulière ≤ 2-3 j) ou sur deux niveaux (story / task tel qu'avec Scrum)
- Une user-story doit être la plus petite possible, mais doit rester d'une taille suffisante pour être significative pour les parties prenantes. De plus, elle apporte toujours de la valeur métier, alors qu'une tâche ne représente qu'une occupation des ressources.
- Seul le reste à faire doit être considéré comme significatif (David Evans)
- Il est indispensable que l'ensemble de l'équipe travaille à l'élaboration du taskboard afin de se l'approprier. Sinon, elle ne l'exploitera pas.

La pause déjeuner a été agrémentée de quelques **Ligthning Talks** dont je ne reprendrai ici que les 2 idées les plus marquantes que j'ai eu le temps de capter (je suis arrivé un peu tard).

- Ce n'est pas l'agilité qui apporte la qualité. C'est la recherche de la qualité qui amène à adopter l'agilité.
- Lorsque la vélocité d'une équipe n'est pas constante, les diverses analyses et statistiques n'apportent souvent pas d'aide. Le plus simple et le plus juste revient souvent à se cantonner à des teeshirts sizes.

L'après midi a ensuite repris avec un sujet au titre intéressant : « **des mots, des maux, démo !** » de Caroline D'amour-Nobi (http://fr.linkedin.com/pub/caroline-damour-nobi) et Emilie Franchomme (http://fr.linkedin.com/in/emiliefranchomme). Le contenu l'était autant que le titre.
Il s'agit bien là de discuter de l'ensemble de la pratique de « revue de sprint » et non simplement de la phase de démonstration.

- Cette étape du process « inspect & adapt » dynamise l'équipe.
- Une revue de sprint se prépare (matériel, invitations, scénario préparé, rôles attribués, préparation de scripts de nettoyage de la base, mise à jour des indicateurs, etc)
- Elle doit s'effectuer sans slide. Les copies d'écrans ou screencasts ne sont acceptables que pour impossibilité technique.
- Préparer l'histoire à raconter
- Déroulement du sprint
- Difficultés rencontrées et surmontées
- User stories non livrables
- User stories livrées (besoin, valeur) peut- être sous forme de Story Telling, ou en exploitant les personas.

- La démo peut aussi se faire en mode « Magicien d'Oz » : l'utilisateur utilise directement l'application. S'il n'y a pas besoin de la moindre explication : c'est gagné !
- La démo force les développeurs à finir et à livrer leur travail (même sur une plateforme différente spécifique à la démo). ➔ On ne montre que ce qui est fini et livrable.
- Il ne faut pas non-plus perdre de temps à la préparation de la démo. Un équilibre doit être respecté.
- Il faut faire attention à la « forme » de validation/rejet des features. Cela peut créer une distance entre l'équipe et le PO si les membres subissent cela comme un tribunal. ➔ On cherche avant tout la coopération et le feedback.
- Le DoD doit être clair et factuel.
- La validation doit se cantonner aux critères d'acceptance définis par le PO *en début* de sprint.
- Le FEEDBACK est le principal but de la revue de sprint.
- Il faut avoir le courage de demander du feedback à *tout le monde*, quelque soit son poste.
    - Ne jamais être sur la défensive.
    - Positiver : Exprimer ce que l'on veut plutôt que ce que l'on ne veut pas.
    - Demander une reformulation et remercier celui qui s'y prête.
    - Inviter des personnes qui ne côtoient pas l'équipe au quotidien.
- La revue de sprint permet d'aligner les équipes avec la vision (qui peut elle-même évoluer).
- Un sprint doit être considéré comme un « petit projet ». La revue de sprint cherche aussi à dynamiser l'équipe en célébrant de petites victoires régulières.

La session suivante était animée par Grégory Alexandre (http://www.coactiv.fr/qui-sommes-nous/l%C3%A9quipe/gregory-alexandre.aspx) qui est responsable QA et Aurélien Fauches (http://www.theses.fr/s35439) qui est doctorant en sociologie religieuse … Et vous me direz : qu'ont-ils bien pu présenter ? Une session intitulée « **Comment vendre le serious game dans votre projet ?** » : prometteur non ?
Bon autant dire que j'en suis ressorti frustré (mais c'est le risque lorsque l'on attend quelque chose de précis).
Donc à défaut de réponse directe à la question qui nous intéressait tous, nous avons eu quelques éléments de réflexion, parfois surprenants (vu la spécialité de l'un des orateurs, la surprise devient finalement prévisible) mais non dénués d'intérêt. Bref, il s'agit d'une analyse sur la pratique du jeu et de ce qu'elle peut apporter et non une clef pour obtenir l'ouverture d'esprit nécessaire à la participation à ses jeux pourtant bien sérieux. Quoi qu'il en soit, voici quelques éléments relevés :

- La pratique du jeu repose sur un respect de la proposition, sur l'ouverture d'esprit des participants, sur la confiance accordée le temps de la découverte.
- Il existe 2 typologies de jeux :
    - Le jeu pour faire avec un objectif de production assistée par un cadre ludique et facilitant la créativité.
    - Le jeu pour apprendre qui permet d'intégrer des connaissances, de tester une situation dans un cadre protégé afin de pouvoir l'appliquer ultérieurement à des fins professionnelles.
- Une référence de jeu que je ne connaissais pas pour découvrir les pratiques Scrum : le Kapla World Tour (http://blog.soat.fr/2012/06/agile-france-kapla-world-tour-revisitez-scrum/)
- Le débriefing est la phase la plus importante du jeu. Elle permet une corrélation avec la réalité et conforte les participants dans le ressenti d'avoir appris quelque chose.
- Il existe des jeux pour tout, mais il faut les choisir en fonction
    - de l'objectif
    - de la culture du public visé
- Pour faire découvrir la pratique de serious games, il vaut mieux commencer par un « jeu pour faire » (planning poker http://planningpoker.com/, consensus workshop http://www.agile-it.fr/post/2012/09/18/Consensus-Workshop-pour-la-retrospective.aspx, product box http://innovationgames.com/product-box/, etc).
- Le « jeu » doit être présenté comme un « atelier ». La notion de jeu a par nature une connotation « peu sérieuse » dans l'inconscient collectif (pour le moment).
- Une session doit être présentée avec un programme cadré, régulé et avec des résultats attendus. Il est cependant parfois difficile d'afficher en début de jeu les résultats attendus alors que le jeu sert souvent à les faire découvrir.
- Il faut chercher du soutien (parfois hiérarchique) dans cette démarche

J'ai ensuite enchainé par une session que je n'étais pas le seul à convoiter (nous étions à l'étroit). Nommée « Kanban, un tour d'horizon de la méthode » et animée par Laurent Morisseau (http://www.laurentmorisseau.com/) auteur du premier livre en français sur la méthode : «Kanban pour l'IT - Une nouvelle méthode pour améliorer les processus de développement » (http://www.amazon.fr/Kanban-pour-lIT-am%C3%A9liorer-d%C3%A9veloppement/dp/2100578677). Il s'agit donc d'une restitution résumée et extrêmement synthétisée de son ouvrage. Pour l'avoir lu, je vous conseille ce livre, et pour y avoir assisté, je vous conseille de réserver une place sur ses prochaines conférences.

En attendant, vous pouvez en avoir un aperçu à travers les slides qui sont déjà disponibles (http://fr.slideshare.net/morisseau/kanban-un-tour-dhorizon-v20). Puisqu'ils sont plus complets que mes notes, je me contenterai de restituer le point qui m'est apparu comme le plus important de la session :
« Ne pas chercher la perfection. Commencer au plus vite pour lancer un mouvement vers le changement ».

J'ai fini ma journée par une présentation surprenante et très agréable. Isabel Monville (http://ayeba.fr/ayeba/equipe/isabel-monville/) nous y a présenté quelques bases de l'analyse transactionnelle (http://fr.wikipedia.org/wiki/Analyse_transactionnelle) dans une session nommée « La communication au service du projet ». J'y ai découvert beaucoup (*en fait tout … j'ignorais jusqu'à l'existence de cette discipline*). L'analyse transactionnelle considère les échanges à travers les états du « Moi » (Parents, Adultes, Enfants) dont le respect permet (entre autres) de ne pas rompre une conversation, et se repose sur des « drivers » (comportements refuges en situation de stress) à identifier chez ses interlocuteurs pour cibler ce qui pourrait entraver cette conversation.
Même si cela paraît évident, pour ne jamais m'être penché sur la question, j'y ai aussi découvert que nous avons tous besoin de reconnaissance. C'est la raison d'être des « strokes » (des signes de reconnaissance). Pour peu qu'ils soient sincères, n'en soyez pas avare, c'est une source de motivation d'une puissance inversement proportionnelle à son coût … et cela ne coûte rien.
Bref, je ne saurais être exhaustif et j'espère ne pas m'être trop éloigné des principes présentés, mais les slides étant déjà en ligne (http://fr.slideshare.net/isabelmonville/la-communication-au-service-du-projet) vous pouvez vous faire une première idée. Espérons que la vidéo ne tardera pas à être mise à disposition.

Voici mon tour d'horizon dont j'ai tiré quelques pistes d'enrichissement de ma bibliothèque.
Prochaine conférence : l'Agile Tour de Toulouse le 25 octobre … à suivre !