

Xtrem Reading



Durée
2h



Préparation
15 minutes



Facilitateur
Optionnel

Résumé

Parcourir à plusieurs une ensemble de livres techniques, pour partager des résumés et si possible trouver un livre à lire pour chaque participant

Préparation

Rassembler des livres intéressants, un par participant. (possibilité de voir à la bibliothèque du Sky)

Déroulé

Le détail des étapes est disponible dans [cet article](#). Il est préférable de faire l'atelier en une seule session (prévoir 2h)

Quand le faire ?

Tous les 3/6/12 mois. Le temps que chacun puisse lire les livres qui l'intéresse et ne pas épuiser trop vite le stock de livres.

Pourquoi c'est bien ?

- Permet de découvrir des livres/notions inconnus
- Encourage les personnes à lire des livres tech

Pourquoi c'est moins bien ?

- Risque que certains livres ne semblent pas utiles à tout le monde
- Survoler les livres peut frustrer certaines personnes

Lean coffee



Durée
30min +



Préparation
5 minutes



Facilitateur
Recommandé

Résumé

Aborder plusieurs sujets en un temps contraint en sélectionnant les plus intéressants pour le groupes. Il est possible de proposer une thématique pour les sujets (nouvelles libs, problèmes récurrents,...)

Préparation

Créer un board. En distanciel, template miro disponible, ou rassembler quelques post-it en physique

Déroulé

Le détail des étapes est disponible dans [cet article](#)

Quand le faire ?

Toutes les semaines/sprints/mois. Attention à la lassitude si on enchaîne trop de fois ce format

Pourquoi c'est bien ?

- Permet d'aborder de nombreux sujets

Pourquoi c'est moins bien ?

- Besoin d'une bonne animation et gestion du temps pour être efficace

Kata



Durée
1h+



Préparation
15 minutes



Facilitateur
Optionnel

Résumé

Pratiquer la résolution de problème, un concept, une méthode... Via un exercice offrant un contexte simplifié par rapport à une application réelle. Un même kata peut être refait plusieurs fois, pour tenter d'autres approches

Préparation

Choisir un kata, initier un repo. [kata-logs](#) offre un tri par thématique (tdd, refacto,...). Vous pouvez aussi en trouver sur [codingdojo](#) ou passer sur [cyber-dojo](#) pour ne pas avoir à créer de repo

Déroulé

En faisant du mob, résoudre le kata et avancer sur les éléments intéressants. Pas de bonne réponse à un kata, l'intéressant est ce qu'en retire les participants

Quand le faire ?

Toutes les semaines/sprints. Faire de la pratique sur du vrai code de temps en temps pour éviter de se lasser des katas

Pourquoi c'est bien ?

- Permet de partager des connaissances et des pratiques
- Permet de sortir de sa zone de confort (kata sur une pratique inconnue, si on s'impose des [contraintes](#), ...)

Pourquoi c'est moins bien ?

- Même points négatifs que le mob programming
- Embêtant si on ne trouve rien d'appliquable dans notre code réel

Learning Hour



Durée
1h-1h30



Préparation
30 minutes



Facilitateur
Optionnel

Résumé

Les learning hours (LH) sont un format issu du [Samman coaching](#), suivant la [structure des 4C](#). L'objectif est de voir un élément précis en une heure (souvent un peu plus). Une [présentation interne](#) existe.

Préparation

Choisir une LH par thème sur le site du samman coaching ou le [repo des coaches](#). Voir le déroulé de la session (celles des coaches on généralement des guides de facilitation)

Déroulé

Enchaîner les 4 étapes (connect, concept, concret practice, conclusion) de l'atelier, en respectant les timings. Généralement un kata est survolé, il est possible de le continuer une fois la LH terminée

Quand le faire ?

Tous les mois/trimestre. Le format demande trop de préparation / connaissances initiales pour être fréquemment joué

Pourquoi c'est bien ?

- Très efficace pour entrer dans un sujet précis
- Structure 4C assure la bonne intégration des concepts

Pourquoi c'est moins bien ?

- Compliqué de jouer une LH sans connaissances initiales sur le sujet, au moins pour l'animateur

Mob Code Review



Durée
20min +



Facilitateur
Inutile



Préparation
5 minutes

Résumé

Faire des code review avec l'ensemble de l'équipe.

Préparation

S'assurer que des demandes de review sont en attente

Déroulé

Le détail des étapes est disponible dans [cet article](#)

Quand le faire ?

Toutes les semaines/sprints/mois. Réduire la fréquence quand moins de discussions ressortent des reviews (équipe mature)

Pourquoi c'est bien ?

- Partage de connaissance et de pratiques
- Permet de faire émerger des standards de code et des checklist pour les reviews

Pourquoi c'est moins bien ?

- Perd de l'intérêt si on ne construit pas de standard et revoit toujours les mêmes problèmes
- Efficace seulement si tout le monde contribue

Analyse post mortem



Durée
20min +



Facilitateur
Inutile



Préparation
5 minutes

Résumé

Analyser un bug / incident résolu, afin de voir si la solution est améliorable, et comment adapter les pratiques de l'équipe pour éviter qu'un problème similaire survienne à nouveau

Préparation

Trouver un bug / incident intéressant

Déroulé

Appliquer les [5 pourquoi](#) sur les questions applicables:

- Pourquoi ce bug a-t-il été écrit ?
- Pourquoi ce bug a-t-il passé les tests ?
- Pourquoi ce bug a-t-il passé la code review ?
- Pourquoi cet incident a-t-il mis du temps à être détecté ?
- Pourquoi le contournement mis en place n'est pas satisfaisant ?

Pour chacune des causes profondes, identifier les actions à mettre en place et les tracer dans un backlog technique ou les résoudre en séance via du mob programming

Quand le faire ?

Tous les sprints/mois/trimestres, plus fréquent si l'équipe a beaucoup de run

Pourquoi c'est bien ?

- Améliore les pratiques/standards/tests de l'équipe

Pourquoi c'est moins bien ?

- Inutile si les actions identifiées ne sont pas mises en place

Mob Programming



Durée
30min+



Facilitateur
Optionnel



Préparation
5 minutes

Résumé

Pratiquer le mob programming sur une US, une tâche ou encore un bugfix

Préparation

Choisir un sujet et créer le [mobtime](#)

Déroulé

Coder la solution au sujet traité, en tournant fréquemment (5-10 minutes) avec [driver/navigateur](#). D'[autres rôles](#) peuvent être utilisés

Quand le faire ?

Toutes les semaines/sprints, tant qu'il y a des sujets. Pour découvrir ensemble un asset mal maîtrisé, une US complexe, ...

Pourquoi c'est bien ?

- Permet de partager des connaissances et des pratiques
- Tout le monde participe

Pourquoi c'est moins bien ?

- Perds beaucoup de son intérêt si on regarde une personne coder pendant une heure sans tourner
- Beaucoup de soucis peuvent survenir, des [solutions](#) existent

Coding games



Durée
30min +



Facilitateur
Inutile



Préparation
5 minutes

Résumé

Résoudre un problème algorithmique en pair ou mob programming

Préparation

Créer un compte sur un site de défis comme [codinggame](#) ou [codewars](#)

Déroulé

Résoudre un problème sur une problématique, maîtrisée ou non, (recherche, structure de données, optimisation, ...). Se mettre en mob ou en pair, de préférence avec au moins un dev expérimenté dans la paire

Quand le faire ?

Toutes les sprints/mois/trimestre. Selon l'appétence de l'équipe et les gains

Pourquoi c'est bien ?

- Permet de partager des connaissances et des pratiques
- Utiles pour revoir les bases si non maîtrisées (structures de données, algorithmes standards, design patterns,...)
- Permet de découvrir des techniques et méthodes du langage, surtout en regardant les meilleures solutions

Pourquoi c'est moins bien ?

- Pas ou peu d'apprentissage si on tombe dans la compétition. Viser le meilleur temps (via un hack ou du code sale) n'est pas le but, mais plutôt de produire un code clair et adapté

Xtrem Watch



Durée
1h30-2h



Facilitateur
Optionnel



Préparation
5 minutes

Résumé

Faire de la veille technique en groupe, et partager le résultat des recherches de chacun

Préparation

Préparer un espace sur miro ou des post-it

Déroulé

Le détail des étapes est disponible dans [cet article](#). L'atelier peut être fait en plusieurs fois mais ce n'est pas l'idéal

Quand le faire ?

Tous les mois/trimestre. Plus fréquemment si la liste des sujets à découvrir / non traités est importante après une première session

Pourquoi c'est bien ?

- Permet de faire une veille efficace rapidement
- Ouvre à des sujets que l'on aurait pas été explorer seul

Pourquoi c'est moins bien ?

- Survol des sujets, cela peut frustrer certains
- Ils est possible que des sujets trop complexes puissent être mal compris en si peu de temps, et que des connaissances incorrectes soient propagées dans l'équipe (jusqu'à ce que quelqu'un se plonge vraiment dans le sujet)

Présentation



Durée
15min +



Facilitateur
Inutile



Préparation
15min +

Résumé

Présentation préparée par un ou plusieurs membres de l'équipe, sur un sujet technique, métier, un asset particulier, ...

Préparation

Le mieux est de préparer un support qui permettra de garder trace des infos partagées (slides, miro, page confluence, ...)

Déroulé

Présenté le sujet aux membres de l'équipe

Quand le faire ?

Tous les mois/trimestre. Plus fréquent risque de devenir ennuyeux

Pourquoi c'est bien ?

- Permet de partager n'importe quel sujet
- Laisse une trace des infos partagées

Pourquoi c'est moins bien ?

- Pas la forme la plus efficace pour apprendre (pas d'interactivité)
- Demande du temps au présentateur(s)
- Si méthode druidique (tradition oral sans écrit, donc pas de trace), la présentation sera vite oubliée et le savoir perdu

Légende



Facilitateur

Note: Les team lead font généralement de très bon facilitateurs

Inutile

Il n'est pas nécessaire d'avoir un facilitateur dédié pour ce format

Optionnel

Un facilitateur dédié serait utile dans ce cas, mais un membre de l'équipe identifié comme facilitateur peut suffire à l'animation

Recommandé

Il est fortement recommandé d'avoir un facilitateur dédié, qui ne soit pas un participant, pour animer sereinement ce format