

Conseils pour bien préparer son hackathon autour d'un projet libre

L'équipe d'OpenHatch organise régulièrement des rencontres, ateliers, événements, sprints, hackathons, etc. invitant de nouveaux contributeurs à participer au développement de logiciels libres. Elle nous livre ici le fruit de son expérience.

L'enthousiasme et la motivation sont indispensables mais ne peuvent faire l'économie d'une solide organisation et réflexion en amont. Happy Hacking ☐

Note : On remarquera que les deux livres cités en référence pour aller plus loin font partie de nos traductions Framabook : Libres conseils et Produire du logiciel libre.



Le guide de l'événement libre in situ

The In-Person Event Handbook

L'équipe OpenHatch – février 2014

(Traduction : Omegax, zak, François, Ju, MrTino, Wan, Asta, François, goofy, amha)

Faire en sorte que votre projet soit prêt pour de nouveaux contributeurs

Il semble que chaque jour apparaisse un nouvel atelier, hackaton ou autre sprint, où des projets open source sont invités à travailler avec de nouveaux contributeurs. À OpenHatch, nous avons nous-mêmes souvent organisé ce genre d'événements ! Nous avons découvert que pour en tirer un profit maximal, il est important de planifier les choses en amont. Expliquer vos objectifs, identifier les tâches appropriées, et tester l'organisation de votre projet sont autant de points indispensables pour aboutir à de belles réalisations et passer un bon moment. Ce travail a grandement amélioré nos expériences, et nous pensons qu'il mérite l'effort significatif qu'il implique.

Nous avons créé ce guide pour aider les projets open source à se préparer pour ces événements. Nous avons utilisé notre propre projet, l'application web OpenHatch.org, comme un exemple dans ce qui suit. Au bas de la page, vous trouverez des aide-mémoires. Elles contiennent en condensé les conseils dispensés dans ce guide, et peuvent vous aider à monitorer vos progrès durant la phase de préparation de votre projet.

Les sources de ce guide sont libres. Un tas de merci à nos contributeurs. (Vous pouvez contribuer, vous aussi !)

Définir des objectifs

Vous devez être capable de présenter clairement vos objectifs pour l'événement, car cela donne à votre groupe une ambition

générale à atteindre. Vous pouvez donc commencer par vous demander :

Quel est l'objectif global de votre projet ?

Il vous faut une réponse courte (un paragraphe ou moins) à cette question que vous pourrez utiliser pour attirer les contributeurs potentiels vers votre projet. Les détails, c'est bien joli, mais vous ne devriez pas avoir besoin d'être trop technique à ce moment là. À de nombreux événements, comme les sprints PyCon, on vous demandera un court résumé à exposer devant tout le monde. Pourquoi ne pas vous y préparer ?

Exemple :

L'objectif de OpenHatch est de rendre les communautés de logiciels libres plus accueillantes pour les nouveaux venus. Pour ce faire, nous fournissons documents et support logistique pour animer des ateliers « Introduction à l'open source », un site web avec des ressources libres, des « missions d'entraînement », un découvreur de talents parmi les bénévoles, et plusieurs autres projets en cours de réalisation.

Que voulez-vous accomplir à cet événement ?

Réfléchissez à ce que vous souhaitez voir spécifiquement réalisé lors de cet événement. Vous pouvez catégoriser ces objectifs selon les différentes parties du projet, si vous en avez plusieurs. Il devrait être spécifié clairement de quelle manière ces actions contribueront au progrès général du projet. Toutefois, il ne s'agit pas de « tâches » – il devrait être toujours nécessaire de diviser ces objectifs pour mieux les atteindre.

Il est utile de les lister en termes d'objectifs « principaux » et « étendus ». Avoir des objectifs principaux réalistes et bien définis vous donne quelque chose à célébrer à la fin de l'événement, tandis que les objectifs étendus ou secondaires vous permettent de prévoir le cas excitant où vous

vous retrouviez avec une équipe nombreuse et efficace, capable d'accomplir une tonne de travail.

En général, il vaut mieux avoir trop d'objectifs que pas assez, mais priorisez-les. Lorsque vous arriverez à l'étape du découpage en tâches, prenez le temps de traiter en détail chaque objectif avant de passer au suivant.

Exemple :

- Faire une nouvelle mission d'entraînement
 - Objectif principal : Choisir une compétence sur laquelle baser une nouvelle mission d'entraînement, et concevoir cette mission. En créer une maquette.
 - Objectif étendu : Implémenter la maquette et la faire tester par des volontaires de l'événement.
- Faire le ménage dans le suivi de tickets
 - Objectif principal : Passer en revue l'outil de suivi, et étiqueter les tickets selon le « ménage » à y faire. Un bug doit-il être vérifié ? Un patch doit-il être testé ? Une demande de fonctionnalité doit-elle être rattachée à un jalon ?
 - Objectif étendu : Utiliser ces étiquettes comme guide pour « faire le ménage » . Vérifier les bugs, tester les patches, etc.

Installation du projet

Dans notre expérience, la phase d'installation du projet est l'obstacle majeur à la participation. Nous avons vu (et conduit !) des événements où les participants passaient le plus clair de leur temps à seulement mettre en place leur environnement de développement et faire connaissance avec le projet. Si notre objectif est de permettre à de nouveaux venus de contribuer, tâchez d'estimer le temps qu'il faut pour installer votre projet. Puis trouvez un ami qui n'est pas

familier avec votre projet pour voir combien de temps il faut vraiment. Vous pouvez aussi trouver quelqu'un pour vous aider à faire cela dans #openhatch.

Documenter et améliorer le processus à l'avance peut faire économiser du temps et de l'énergie à tout le monde. Si vous savez qu'une partie de votre projet sera inévitablement dévoreuse de temps, assurez-vous que tous les participants savent exactement à quoi s'en tenir.

Toutes les informations ci-dessous devraient être documentées dans un fichier LISEZ-MOI placé à la racine de votre dépôt. Vous pouvez également placer cette information dans une section « Vous voulez participer ? » sur le site web de votre projet, et/ou vous pouvez inclure un lien vers le LISEZ-MOI dans la signature de votre liste de diffusion ou dans la barre de statut de votre canal IRC.

Comment trouver une communauté et des mainteneurs pour le projet

Les informations des contacts devraient être explicitement affichées, étant donné que vous pourriez avoir des contributeurs éloignés ou des contributeurs qui voudraient démarrer en amont de l'événement. Les types d'informations de contacts peuvent inclure :

- Un lien vers votre mailing-list ;
- Votre nom de canal et de serveur IRC (avec le lien vers le guide pour installer IRC et le lien vers sa version webchat) ;
- Les comptes de réseaux sociaux tels que Identi.ca, Twitter ou Facebook si votre projet en possède ;
- Le contact des mainteneurs du projet, si vous vous sentez à l'aise pour le mentionner.

Si vous avez une préférence pour le type de prise de contact, précisez-le.

Exemple:

OpenHatch laisse en deux endroits des informations de contact que nous essayons le plus possible de garder à jour et en cohérence l'un avec l'autre. Il y a nos informations de contacts dans la documentation, référencées principalement depuis notre répertoire où sont déposés les codes sources, et nos informations de contacts dans le wiki, référencées à partir de la page d'accueil du site web.

La structure du projet

Décrivez la structure de base de votre projet. Quels sont les plus gros composants et où sont-ils situés ? Comment ces composants interagissent-ils ? Puis décomposez chaque partie. Vous n'avez pas à parler de chaque fichier ou sous-dossier de votre projet, mais prenez soin de ne pas prendre pour acquis que chaque nouveau venu comprendra le sens de tel script, ou la manière dont interagissent tels fichiers, ou dans quel langage est programmée telle partie du projet.

Selon la taille et la complexité de votre projet, cela peut être une tâche particulièrement imposante. Chez OpenHatch, nous travaillons toujours à documenter la structure complète du projet. Nous recommandons de commencer par une explication « en vue d'ensemble » de la structure projet – juste assez de détails pour remplir d'une demi-page à une page. Quand vous aurez plus de temps, vous pourrez rentrer dans le détail, en commençant par les parties du projet sur lesquelles les gens travaillent le plus souvent (ou qui sont plus susceptibles de faire l'objet de sprints ou de hackatons). Si vous utilisez d'autres frameworks ou bibliothèques, vous pouvez gagner du temps en mettant des liens vers leur documentation et leurs tutoriels.

Exemple :

Une description d'ensemble de la structure du projet OpenHatch peut être trouvée dans « Project Overview ». Une description

de la structure de OH-Mainline (le dépôt derrière notre site web) y est présente.

Comment mettre en place un environnement (« de développement ») local

Pour contribuer à votre projet, les gens ont généralement besoin d'une version locale du projet où ils peuvent faire des modifications et les évaluer. Plus votre guide d'installation/développement est détaillé et clair, mieux c'est.

Voici quelques éléments pour la mise en place d'un environnement de développement à mettre dans votre guide :

- Préparer leur ordinateur
 - Assurez vous qu'ils connaissent bien les outils de leurs systèmes d'exploitation, comme le terminal / la ligne de commande. Vous pouvez le faire en donnant un lien vers un didacticiel et en demandant aux utilisateurs s'ils le comprennent bien. Souvent, de très bons didacticiels existent déjà, celui d'OpenHatch sur la ligne de commande est disponible [ici](#).
 - Si les contributeurs doivent mettre en place un environnement virtualisé, accéder à une machine virtuelle ou installer un kit de développement en particulier, expliquez-leur comment faire.
 - Donnez la liste de toutes les dépendances nécessaires pour faire tourner le projet, et une explication pour les installer. Fournissez un lien vers de bons guides d'installation, s'il en existe.
- Téléchargement du code source
 - Donnez des instructions détaillées sur comment télécharger le code source du projet, dont les obstacles et problèmes fréquemment rencontrés.
 - S'il existe plusieurs versions du projet, dites

clairement quelle version ils doivent télécharger.

- Comment voir / tester les changements
 - Expliquer aux contributeurs comment voir et tester les changements qu'ils ont effectués. Ceci peut varier selon ce qu'ils ont changé, mais faites votre possible pour couvrir les changements les plus fréquents. Cela peut être, tout simplement, afficher un document html dans un navigateur, mais ce peut aussi être plus compliqué.

L'installation pourra différer pour chaque contributeur en fonction de leur système d'exploitation. Vous aurez probablement besoin de créer des instructions séparées pour les utilisateurs de Windows, Mac et Linux, dans différentes parties de votre guide. Si vous entendez ne supporter le développement que pour un seul système d'exploitation, assurez-vous que cela soit dit clairement pour les utilisateurs,

Exemple:

Vous pouvez voir comment OpenHatch annonce cela dans son guide d'installation. Les instructions pour tester des changements peuvent être trouvées là. Des tâches plus spécifiques peuvent avoir leur documentation additionnelle (par exemple la documentation concernant les différents changements du code).

Contributions et retours

Comment les contributeurs doivent-ils procéder pour amener leurs modifications au projet ? Est-ce qu'ils soumettent une *pull request* sur Github ? Doivent-ils générer un patch et l'attacher à un ticket sur le système de tickets ? Assurez-vous que cette information est explicitement fournie.

Exemple :

Le guide d'OpenHatch pour soumettre des changements peut être trouvé [ici](#).

Il est aussi utile d'indiquer aux gens comment ils peuvent faire des retours/signaler des bugs sur le projet. Si votre projet n'a pas de système de suivi de tickets, envisagez d'en créer un. Sur Github, tous les dépôts disposent du système de tickets (bien que vous ayez à l'activer en allant dans Settings puis Features). Il y a beaucoup d'autres systèmes de suivi de tickets.

Si votre projet est de petite taille, vous pouvez ne pas avoir besoin ou ne pas vouloir de systèmes de suivi de tickets. Aucun problème. Le principal est que les contributeurs sachent comment vous remonter des informations.

Exemple :

Les problèmes avec le projet « Open Source Comes to Campus » peuvent être signalés [ici](#). La plupart des autres problèmes liés à OpenHatch peuvent être signalés [là](#).

Les outils tels que le suivi de tickets sont très utiles pour communiquer de manière asynchrone. Ce n'est peut-être pas la solution la plus appropriée lors d'un événement physique. Si vous voulez changer les choses pour l'occasion – comme demander aux participants de vous notifier par IRC les liens vers les nouveaux tickets, pour éviter qu'ils ne passent entre les mailles du filet – assurez-vous de leur en parler !

Vérifiez votre documentation

Vérifiez que cette documentation est complète et effective en la testant auprès de personnes qui n'ont pas participé à votre projet auparavant. Pour chaque système d'exploitation, trouvez au moins une personne pour lire votre documentation, tenter d'installer, faire et tester des modifications et participer aux différentes modifications du projet (au choix de votre testeur, ces modifications peuvent être des faux changements ou des vraies tâches). Vérifiez que vos testeurs ont des compétences et/ou une expérience similaire à celle des nouveaux arrivants à votre événement.

Si vous rencontrez des difficultés pour trouver des personnes pour vous aider, essayez la chaîne #openhatch sur IRC.

Assurez-vous que tous les problèmes qui surviennent lors de ce processus de vérification soient intégrés à la documentation. Une fois que la documentation a été vérifiée, ajoutez une ligne au début de votre guide pour annoncer ce qui a été vérifié et quand.

Par exemple :

Instructions de l'environnement de développement testées avec succès sur Ubuntu 12.04 (le 03/10/2013), Mac OS X 10.8 (le 01/10/2013) et Windows XP (en janv. 2005). Vous pouvez consulter cette démarche à OpenHatch [ici](#).

Idéalement, vous devriez vérifier que tout fonctionne : installer, faire des modifications, les tester et les intégrer. Si vous n'avez le temps que pour une seule de ces tâches, nous vous recommandons de vérifier l'installation. Notre expérience nous a appris que c'est sur ce point qu'émergent la plupart des problèmes.

Définir les tâches des participants

Retournons aux objectifs de l'événement dont nous avons parlé dans la première section. Chaque objectif devrait être décomposé en étapes distinctes qui permettront de l'atteindre. Ces étapes sont les tâches à confier aux participants.

Ces tâches devraient inclure un résumé en *anglais simple* aussi bien que les informations sur la localisation des modifications (par exemple, quels fichiers ou fonctions sont altérés). Nous recommandons d'inclure une liste des compétences nécessaires (par exemple : *compétences en design, Python basique*) et des outils (par exemple: *Développement sur environnement Mac*). Il est aussi utile d'inclure le nombre estimé d'heures que la tâche va prendre, d'étiqueter certaines tâches comme plus ou moins importantes, et de marquer où

quelle tâche est dépendante d'une autre.

Cela semble représenter beaucoup de travail mais cela devrait aider vos participants à trouver rapidement et facilement les tâches appropriées pour eux. Etant donné que l'un des objectifs principaux d'un événement *en personne* est de donner une expérience positive aux participants, nous pensons que cela en vaut la peine.

Créer un système pour suivre les tâches

Nous recommandons l'utilisation d'un wiki ou un document de planification similaire pour garder une trace des tâches. OpenHatch dispose d'un navigateur pour suivre les tâches que nous utilisons pour nos événements. Vous pouvez, si vous le souhaitez, faire un *fork* et l'adapter à votre projet ou événement ; cependant, nous vous recommandons d'attendre un peu, car nous allons bientôt faire de grosses améliorations. Quelque chose de très simple, comme un etherpad (*NdT : ou Framapad*), peut également faire l'affaire (ici, un cadre et un service exemple que vous pouvez utiliser).

Préparer les tâches

Pour vous rendre compte du nombre de tâches à préparer, nous vous recommandons de vous baser sur la durée de l'événement et le nombre de participants attendus pour prédire la charge en temps/homme de votre projet. Vous pouvez utiliser les estimations de temps que vous avez faites pour évaluer chaque tâche. Nous suggérons que vous prévoyiez 30% de plus que ce dont vous pensez que vous aurez besoin : il vaut mieux avoir trop que pas assez.

Exemple :

- Objectif de base : parcourir l'interface et étiqueter les tickets selon le type de résolution nécessaire. Est-ce qu'un bug doit être vérifié ? Est-ce qu'un patch doit être testé ? Une fonctionnalité doit-elle être rattachée à un jalon ?

- Tâche 1 : étiqueter les problèmes
 - Compétences et outils nécessaires : compétences de base en anglais, connaissance des concepts de vérification, tests, jalons...
 - Temps estimé : ~20 minutes par problème
 - Pour démarrer : prenez connaissance du suiveur de tickets, et comment il affiche les informations. (Voyez cette documentation.) Demandez un accès *Administrateur* pour pouvoir ajouter des étiquettes au suiveur.
 - Pour chaque problème : lisez la discussion relative à chaque problème, et identifiez où la communauté intervient dans le signalement de ce problème. Si c'est un bug non vérifié, ajoutez le label *non vérifié*. Si c'est un patch non testé, ajoutez l'étiquette *patch non testé*, etc. Si c'est une fonctionnalité qui n'est rattaché à aucun jalon, ajoutez l'étiquette *jalon nécessaire*.
- Objectif étendu : utilisez les étiquettes comme un guide pour résoudre chaque ticket. Vérifiez les bugs, testez les patch, etc.
 - Tâche 1 : Vérifier les bugs
 - Compétences et outils nécessaires : compétences de base en anglais, et dans l'idéal connaître les machines virtuelles pour effectuer des tests sur plusieurs OS.
 - Temps estimé : 20 minutes par bug (variance élevée).
 - Pour démarrer : téléchargez l'environnement de développement et assurez vous que vous pouvez lancer le projet. Assurez vous que vous avez un compte sur *Le suiveur de tickets* et que vous savez ajouter des commentaires ou changer les étiquettes.
 - Pour chaque bug : essayez de reproduire le

bug. Enregistrez les résultats dans un commentaire, avec des informations concernant le type de l'OS et sa version. Si possible, effectuez le test sur plusieurs navigateurs. S'il y a plusieurs commentaires récents sur les trois OS majoritaires, ajoutez l'étiquette *prêt pour test par le mainteneur* (NdT : *ready_for_maintainer_review*).

Quoi qu'il en soit, les participants doivent être rattachés à une tâche correspondant à leurs compétences et à leurs intérêts. Effectuer ce travail préparatoire fera démarrer les participants immédiatement plutôt que de les faire attendre que vous leur suggériez une tâche adéquate. Dans l'idéal, les organisateurs d'événements auront collecté des informations sur les compétences et intérêts des participants bien en amont, donc vous pourrez adapter la liste à votre groupe de contributeurs.

Rendre explicites les étapes de chaque tâche facilite l'entraide entre les participants. En identifiant clairement les compétences et concepts nécessaires, il devient plus facile pour les gens de dire : « Oh, je comprends comment faire ça ! Attends, je vais te montrer ».

Suite

Il est possible que les contributeurs ne puissent pas finir les tâches sur lesquelles ils travaillent pendant l'événement, ou qu'ils veuillent continuer à participer au projet en travaillant sur d'autres tâches. Penser en amont de l'événement à comment vous allez organiser sa suite rend plus facile l'échange d'informations avec les participants et la planification de la direction de votre projet.

Nous recommandons de demander à chaque participant de répondre aux questions suivantes à propos des tâches sur lesquelles ils

ont travaillé. Donnez leur cette liste au début de l'événement : cela les aidera à documenter ce qu'ils font. Vous pouvez imprimer cette liste, l'envoyer par mail aux participants, faire un formulaire web, comme vous préférez.

Par exemple :

- Pour chaque tâche sur laquelle vous avez travaillé, veuillez répondre aux questions suivantes :
 - Sur quelle tâche avez-vous travaillé ?
 - Veuillez documenter brièvement votre *workflow*. Quelles étapes, dans quel ordre, pourquoi ?
 - Où puis-je trouver le travail que vous avez réalisé pendant l'événement ? Ceci inclut le code, la documentation, les maquettes, et autres choses.
 - Si vous avez créé des comptes pour le projet, veuillez lister les sites et les noms des comptes. Enregistrez le mot de passe dans votre gestionnaire de mot de passes préféré, ou assurez vous que moi ou un autre mainteneur y aie accès.
 - Quels obstacles avez-vous rencontré en travaillant sur cette tâche ? Avez-vous des retours à me faire pour améliorer le processus pour les futurs contributeurs ?
 - Voudriez-vous rester impliqué dans ce projet ? Si oui, dans quelle mesure ?

S'il y a suffisamment d'enthousiasme pour continuer le travail, assurez-vous de rester en contact ! Nous suggérons que vous rassembliez les emails des participants intéressés et que vous les contactiez dans les 48 heures suivant l'événement. Dans votre email, remerciez-les pour leur aide et fournissez des informations sur comment rester dans la communauté par, par exemple, IRC ou des listes de diffusion.

Nous recommandons aussi de prévoir un rendez-vous suite à l'événement. Si vous êtes tous originaires de la même région, essayez de fixer une date après l'événement pour vous et votre

équipe au café du coin, à l'espace de coworking, ou à une soirée-projet. Si vous êtes éloignés géographiquement, fixez une date pour vous rencontrer sur IRC ou sur Google Hangout. 2-3 semaines après l'événement est un bon créneau, toutefois cela dépend si vous et vous nouveaux contributeurs êtes très occupés.

Checklists

Merci de votre attention ! Pour vous aider à garder un œil sur chaque étape, nous avons créé deux checklists pour vous. La version détaillée inclut tous les conseils ci-dessus. La checklist la plus courte inclut les conseils que nous pensons les plus importants. Nous vous recommandons de démarrer avec la checklist la plus courte. Une fois que vous avez accompli correctement cette checklist, vous pouvez revenir et faire les étapes supplémentaires si vous avez le temps et l'énergie.

Pour voir et/ou imprimer les checklists, se rendre [ici](#).

Remerciements

Merci à tous ceux qui ont contribué ou aidé au projet.

Contributeurs

- Shauna Gordon-McKeon : maintenance, contenu
- Ni Mu : design
- Sheila Miguez : testeuse
- Asheesh Laroia : testeuse

Pour aller plus loin

- Open Advice (*NdT : entièrement traduit par nos soins*)
- Producing Open Source Software (*NdT : entièrement traduit par nos soins*)