

Gérer les versions de son code

Versionner son code signifie garder une trace des différentes étapes de développement d'un projet logiciel. Cela consiste à sauvegarder régulièrement le code sous différentes versions, ce qui permet de revenir à un état antérieur en cas de problème. Même sans utiliser d'outils complexes, versionner son code peut se faire manuellement et reste indispensable, que l'on travaille seul ou en équipe.

5TTR

6TTR

6TQ



Découverte

Pourquoi versionner son code quand on travaille seul ?

1. Protéger contre les erreurs

Quand on développe un projet, il est fréquent de rencontrer des erreurs après avoir effectué des modifications. Si vous avez sauvegardé une version précédente de votre code, il vous est facile de revenir en arrière et de retrouver une version stable. Cela vous évite de perdre du temps à chercher où l'erreur a été introduite.

Exemple pratique : Vous travaillez sur une fonctionnalité de votre site web. Tout fonctionnait bien avant que vous n'ajoutiez une nouvelle fonctionnalité, mais après plusieurs heures, rien ne marche comme prévu. Sans versionnage, vous seriez obligé de corriger le code à tâtons. Avec une sauvegarde précédente, il suffit de récupérer l'ancienne version pour repartir d'une base saine.

2. Suivre l'évolution du projet

Lorsque vous sauvegardez régulièrement votre code, vous créez une sorte d'historique de l'évolution de votre projet. Cela vous permet de suivre votre progression, d'analyser les changements effectués à chaque étape, et de mieux comprendre comment et pourquoi certaines décisions ont été prises.

Exemple pratique : Au début de votre projet, vous avez créé une première version basique d'une application de gestion de tâches. Quelques semaines plus tard, vous avez ajouté des filtres et une interface utilisateur plus moderne. En sauvegardant différentes versions au fil du temps, vous pouvez voir l'évolution du projet et même revenir à une version simplifiée si nécessaire.

3. Faciliter les expérimentations

Versionner son code permet d'expérimenter de nouvelles idées sans crainte. Si une nouvelle fonctionnalité ne fonctionne pas comme prévu, il est toujours possible de revenir à la version précédente. Vous pouvez ainsi tester de nouvelles approches sans avoir peur de casser quelque chose d'important.

Exemple pratique : Vous décidez d'essayer un nouveau design pour votre site web. Vous faites une sauvegarde de la version actuelle, puis vous commencez vos modifications. Si le nouveau design ne vous plaît pas ou crée des problèmes, vous pouvez simplement restaurer la version précédente.

4. Éviter la perte de données

Il est aussi important de versionner son code pour éviter la perte de données en cas de problème matériel ou de suppression accidentelle. En sauvegardant régulièrement votre code, vous vous assurez d'avoir une copie récente et utilisable, même en cas d'incident.

Exemple pratique : Imaginez que votre ordinateur tombe en panne alors que vous n'avez pas sauvegardé votre travail depuis plusieurs jours. Si vous aviez pris l'habitude de sauvegarder régulièrement votre code sur un autre support (clé USB, disque dur externe), vous n'auriez pas tout perdu et pourriez reprendre le travail là où vous l'aviez laissé.

Pourquoi versionner son code quand on travaille en équipe ?

1. Coordination des modifications

Lorsque plusieurs personnes travaillent sur un même projet, il devient essentiel de bien gérer les différentes modifications apportées par chacun. Le versionnement permet à chaque membre de l'équipe de sauvegarder ses modifications sans écraser le travail des autres. Chaque version du projet peut ainsi être comparée, et les modifications peuvent être intégrées progressivement.

Exemple pratique : Imaginez deux développeurs qui travaillent sur une même application. L'un d'entre eux ajoute une nouvelle fonctionnalité tandis que l'autre corrige un bug. Sans versionnage, le second développeur pourrait écraser les changements du premier, créant des conflits et des erreurs. En versionnant le code, chacun peut apporter ses modifications de manière sécurisée.

2. Suivre les contributions individuelles

Versionner son code permet également de garder une trace des contributions de chaque membre de l'équipe. Chaque sauvegarde correspond à une version du projet avec un historique détaillant qui a apporté les changements. Cela permet de mieux comprendre d'où vient un problème ou d'identifier qui a ajouté une fonctionnalité particulière.

Exemple pratique : Si une erreur est introduite dans une fonctionnalité, il est facile de retrouver qui a fait la modification en consultant l'historique des sauvegardes. Cela permet à l'équipe de discuter ensemble pour comprendre l'origine du problème et trouver la meilleure solution.

3. Revenir à une version stable en cas de problème

En travaillant en équipe, il peut arriver que des modifications introduisent des erreurs dans le projet. Avec un système de versionnement, il est facile de revenir à une version stable du code avant que les problèmes n'apparaissent. Cela permet de maintenir un projet fonctionnel tout en corrigeant les erreurs en parallèle.

Exemple pratique : Une équipe de trois développeurs travaille sur une application. Après l'ajout d'une nouvelle fonctionnalité, l'application commence à planter régulièrement. Grâce au versionnage, l'équipe peut revenir à la dernière version stable et travailler sur les correctifs sans interrompre le fonctionnement de l'application.

4. Sauvegarder le travail de chacun

Dans une équipe, chaque développeur peut sauvegarder son travail de manière indépendante. Cela permet à chaque membre d'avoir une copie à jour de l'état du projet, et d'éviter que les changements apportés par un membre ne perturbent les autres.

Exemple pratique : Un développeur commence à travailler sur une fonctionnalité complexe, mais ne la termine pas dans la journée. En versionnant son code, il peut sauvegarder son travail à mi-parcours sans impacter les autres membres de l'équipe, qui continuent de travailler sur d'autres parties du projet.

Bonnes pratiques pour un versionnage manuel

Même sans utiliser d'outils sophistiqués comme Git, il est possible de versionner son code de manière manuelle en suivant quelques bonnes pratiques simples :

- 1. Sauvegarder régulièrement** : Prenez l'habitude de faire des copies de votre projet à des étapes importantes, par exemple avant et après l'ajout d'une fonctionnalité, la correction d'un bug ou des modifications majeures.
- 2. Utiliser une convention de nommage claire** : Donnez des noms explicites à vos versions pour identifier facilement leur contenu. Par exemple, utilisez des noms comme `Projet_v1_ajout_interface` ou `Projet_v2_correction_bugs`.
- 3. Conserver un historique** : Ne supprimez pas les anciennes versions. Gardez un historique de votre projet avec plusieurs versions sauvegardées, afin de pouvoir revenir à un point antérieur si besoin.
- 4. Sauvegarder sur différents supports** : Ne gardez pas toutes vos sauvegardes sur un seul appareil. Utilisez plusieurs supports (comme des disques durs externes ou un stockage en ligne) pour protéger votre projet en cas de panne.

Conclusion

Versionner son code est une étape essentielle dans tout projet de développement, même si l'on travaille seul. Cela permet de suivre l'évolution du projet, de corriger les erreurs plus facilement, et de protéger son travail contre les incidents. En équipe, le versionnage devient encore plus indispensable pour garantir la coordination des membres et la stabilité du projet.

Adopter des méthodes simples de versionnement, comme des sauvegardes manuelles régulières, est un bon point de départ. Dans un prochain article, nous explorerons des outils plus avancés, comme **Git**, qui permettent d'automatiser et d'améliorer le processus de versionnement.

