

Introduction au DevOps

La programmation et le développement

Programmer

Programmer

Écrire une recette de cuisine pour automatiser une fonction

- Des "ingrédients" en entrée (les données fournies par l'utilisateur)
- Des étapes définies pour manipuler ces ingrédients (l'algorithme)
- Un résultat utile en sortie (Les informations et contenus renvoyés à l'utilisateur)
- Cf. Module sur la programmation Java en semaine 2.

Programmer

- L'algorithme est décrit avec un langage strict

Programmer

- L'algorithme est décrit avec un langage strict
- On utilise des fonctions existantes (les bibliothèques et frameworks) pour coder le minimum de choses nouvelles.

Programmer

- L'algorithme est décrit avec un langage strict
- On utilise des fonctions existantes (les bibliothèques et frameworks) pour coder le minimum de choses nouvelles.
- On construit des programmes en assemblant des fonctions.

Programmer

- L'algorithme est décrit avec un langage strict
- On utilise des fonctions existantes (les bibliothèques et frameworks) pour coder le minimum de choses nouvelles.
- On construit des programmes en assemblant des fonctions.
- On compile le code et on installe les dépendances (les briques de base) pour que la machine puisse exécuter le programme.

Le développement

C'est plus que de la programmation

- Il s'agit de **répondre à un besoin**.

Le développement

C'est plus que de la programmation

- Il s'agit de **répondre à un besoin**.
- Le besoin est généralement défini par un client ou un chef produit qui n'est pas dans la tête du développeur

Le développement

C'est plus que de la programmation

- Il s'agit de **répondre à un besoin**.
- Le besoin est généralement défini par un client ou un chef produit qui n'est pas dans la tête du développeur
- Il faut **définir** et redéfinir le besoin client **explicitement**

Le développement

Est un travail d'équipe

Écrire quelque chose de logique et cohérent tout seul c'est beaucoup plus simple que quand on est plusieurs.

- Il faut des outils comme **git** qui permet de **gérer plein de versions différentes** du logiciel à la fois et les **combiner**. Cf module sur l'usine logicielle semaine 3.

Le développement

Est un travail d'équipe

Écrire quelque chose de logique et cohérent tout seul c'est beaucoup plus simple que quand on est plusieurs.

- Il faut des outils comme **git** qui permet de **gérer plein de versions différentes** du logiciel à la fois et les **combiner**. Cf module sur l'usine logicielle semaine 3.
- Il y a plusieurs métiers dans une équipe de dev (développeur, architecte, chef produit, testeur)

Développement Agile

Le problème historiquement

- Pour avoir des logiciels de qualité, le **mouvement de l'ingénierie logicielle** a défini depuis les années 80 pleins de procédures bien réglées pour être méthodique et rien oublier.
- D'un côté c'est une bonne chose : lorsqu'on travaille sur des milliers de lignes de code pendant des années et ça devient rapidement désordonné et ingérable si ce n'est pas organisé.
- D'un autre côté : les procédures sont **ennuyeuses** et entrave le développement **rapide et spontané** du projet.
- Dans ce contexte de nombreux projets logiciels sont terminés avec des mois voire des années de retard et font perdre beaucoup d'argent : ils sont alors déjà dépassés lorsqu'ils sont terminés.

Développement Agile

Pour répondre à ce problème, le mouvement du développement agile propose de changer de principe :

- Revenir à une méthode de développement plus **spontanée** et **artisanale**
- Mais faire des **petites itérations**...
- **Valider à chaque itération** avec le reste de l'équipe et donc le chef produit.

Développement Agile

Petites itérations

- Faire une première version avec juste une ou deux fonctionnalités principales

=> Il faut que ce soit **programmable en quelques semaines** maximum.

- Puis on démarre un autre cycle en ajoutant seulement une ou deux fonctionnalités à la fois par ordre de priorité.

=> on crée alors environ une **nouvelle version** du code **chaque jour** dans une équipe moyenne (20 personnes)

Développement Agile

Aujourd'hui l'Agile est devenu mainstream : presque toutes les entreprises cherchent à fonctionner avec ce **dynamisme**.

C'est très adapté en particulier à l'innovation et au format startup.

Lean Manufacturing

Valable pour les Toyotas mais aussi pour le logiciel.

Lean Manufacturing

Valable pour les Toyotas mais aussi pour le logiciel.

Ne pas anticiper la demande = principe de Traction

On attend qu'une fonctionnalité du logiciel soit nécessaire plutôt que d'avoir un cahier des charges énorme dès le départ.

Lean Manufacturing

Valable pour les Toyotas mais aussi pour le logiciel.

Ne pas anticiper la demande = principe de Traction

On attend qu'une fonctionnalité du logiciel soit nécessaire plutôt que d'avoir un cahier des charges énorme dès le départ.

Travailler en flux tendu

On n'accumule pas de retard et on réagit vite à la demande. Pour ça il faut que le code soit le plus propre possible à chaque étape de la construction du logiciel

Bilan : faire des
cycles court et

État d'esprit du développement

- Livrer un produit conforme (les contraintes viennent du marché et de l'extérieur pas du système)

État d'esprit du développement

- Livrer un produit conforme (les contraintes viennent du marché et de l'extérieur pas du système)
- Ajouter des fonctionnalités pour être compétitif (se développer) -> vitesse

État d'esprit du développement

- Livrer un produit conforme (les contraintes viennent du marché et de l'extérieur pas du système)
- Ajouter des fonctionnalités pour être compétitif (se développer) -> vitesse
- Les détails de l'infrastructure sont secondaires (il faut que ça fonctionne mais ce n'est pas le travail du développeur).

Conclusion

On voit bien que l'état d'esprit du développement et des opérations sont incompatibles voire intrinsèquement opposés.

