Objectif DevOps : Améliorer la collaboration entre les équipes de dev. Livraison de logiciel de haute qualité (rapide). Pratique et philosophie pour accélérer la livraison logicielle.

**Principes de base :**

* Collaborer et communiquer entres les équipes
* Automatisation, des processus
* Intégration et déploiement continus (mise en place d’outils et intégration du processus)
* Feedback et amélioration continue

Relation entre DevOps / Agile / Lean :

DevOps s’appuie sur la méthode Agile et Lean pour les compléter.

Pas spécialement adapté pour toutes les entreprises.

Comprendre le travail de son équipe pour pouvoir intervenir en cas d’absence d’une personne responsable d’une certaine partie du projet.

**Communication et collaboration entre les équipes :**

* Partage des responsabilités et des objectifs communs
* Réunions régulières et rétrospectives pour discuter des progrès des amélioration (Chaos Monkey)
* Pratiques de travail en équipe, telles que le pair prograing (2 devs sur un seul code) et la revue de code.
* Optimisation des échanges (<https://nohello.club/>) (aller droit au but)

**Adoption d’outils et de processus automatisés :**

* Intégration continue et déploiement continue
* Gestion automatisée des infrastructures et des configurations
* Test automatisés et monitoring en temps réel

**Mise en place de métriques et d’indicateurs de performances (KPI) :**

* Délai de mise sur le marché (time-To-Market)
* Taux de réussite des déploiements
* Temps de résolutions des incidents
* Nombre de bogues et de problèmes de performances résolus

**Exemples et résultats de projets DevOps réussis :**

Exemple Netflix :

* Fournir du services 24/24 de haute qualité
* Gérer un déploiement mondial (pas les mêmes horaires pour l’ensemble du monde)
* Les outils automatisés sont utilisés mondialement peu importe les pays.
* 700 déploiements par heure au niveau mondial
* Permet la stabilité et l’amélioration du service ce qui permet d’avoir un service stable et très disponible.

Exemple Etsy (e-commerce):

* Défi : améliorer la collaboration entre les équipes
* Strat : batterie de tests / automatisation des tests
* Résultats : réduction des erreurs et plus grande fréquence de

Exemple Amazon :

* Défi : gérer la croissance rapide (pb de riche (argent))
* Strat : Adopte la culture DevOps, CI/CD, automatisation des tests, micro-services
* Résultats : Stabilité du service, déploiements plus rapides et fréquents

**Analyse des bénéfices obtenus :**

Analyse des problèmes et mettre en place des solutions.

**Rôles et responsabilités**

* Développeur à concevoir et développer, Collaborer avec l’équipe Ops pour automatiser les processus de déploiement
* Ingénieur Ops à Gérer les infrastructures et les configurations, automatiser les processus de déploiement, assurer la disponibilité
* Testeur à Concevoir des automatismes de test / collaborer avec les équipes Dev et Ops
* Product Owner à Définir les priorités et les objectifs du produit, planifier les sprints et gérer les attentes des parties prenantes.
* Scrum Master à Faciliter la communication et la collaboration entre les équipes Dev et Ops, encourage l’amélioration continue et gérer les différents obstacles et problèmes.

**Compétences techniques requises :**

* Langages de programmation et des Frameworks (java, python etc.…)
* Maitrise des outils d’intégration et déploiement continue (Jenkins Gitlabs CI/CD)
* Connaissance des outils d’automatisation des infrastructures (Ansible, chef, Puppet)
* Expérience avec les plateformes de conteneurisation et d’orchestration (Docker, Kubernetes)
* Connaissance des outils de monitoring et de performance (New Relic)
* Expérience en matière de tests automatisés (par exemple, Sélénium, JUnit, TestNG)
* Connaissance des méthodologies Agile et Lean
* Compétences en gestion de projet et en planification

**Positionnement et périmètre du client :**

Analyser les objectifs et les défis de l’entreprise --> Évaluer l’état actuel des processus de développement et d’exploitation --> Identifier les parties prenantes et leurs attentes --> Comprendre les contraintes techniques, budgétaire et temporelles --> Prioriser les besoins en fonctions des objectifs de l’entreprise.

**Adapter les solutions DevOps :**

Améliorer la collaboration

Accélérer le déploiement

Optimiser les infrastructures

Améliorer la qualité

**Collaboration et Communication :**

Organiser des réunions de travail inter-équipe

Utiliser des outils de communication et de collaboration

Créer des espaces de travail partagés pour faciliter la collaboration

Encourager l’entraide et le partage des connaissances entre les membres de l’équipe

Mettre en place des processus de revue de code et de retour d’information

**Gestion des conflits et résolution de problèmes :**

* Adopter une approche centrée sur les problèmes plutôt que sur les personnes
* Ecouter activement et valider les préoccupations des membres de l’équipes
* Encourager l’expression des désaccords et opinions divergentes
* Rechercher des compromis et des solutions gagnant-gagnant
* Utiliser des techniques de résolution des problèmes comme le brainstorming ou l’analyse des causes probables.

**Animation d’une équipe DevOps**

**Techniques d’animation et de motivation :**

* Définir des objectifs clairs et réalisables pour chaque membre de l’équipe
* Reconnaitre et célébrer les réussites les progrès accomplis
* Fournir des opportunités de formation et de développement professionnel
* Organiser des évènements d’équipe et des activités de team-building
* Favoriser un environnement de travail positif et soutenant

**Gestion de la performance :**

* Productivité
* Qualité
* Efficacité

**Culture DevOps :**

* Collaboration
* Communication
* Automatisation
* Mesure
* Amélioration continue

JOUR 2

**Sensibilisation et formation :**

* Organiser des ateliers et des formations sur la philosophie DevOps, ses principes et ses pratiques
* Présenter les bénéfices et les avantages de l’adoption du DevOps pour l’entreprise et les équipes
* Proposer des formations spécifiques aux outils et technologies utilisés dans le cadre des projets DevOps

**Partage des bonnes pratiques :**

* Organiser des réunions
* Créer des espaces de collaboration en ligne
* Encourager les retours d’expérience par les membres de l’équipe
* Favoriser le mentorat et le coaching pour partager les compétences dans l’équipe
* Favoriser une culture d’amélioration / se remettre en question

**L’agilité dans DevOps :**

DevOps est née de l’agilité, permet la rapidité, l’amélioration et la livraison continue + automatisation.

**Principes agiles pour faciliter la transition vers DevOps :**

* DevOps s’appuie sur la méthode agile, il ne remplace pas il étend.
* Encourager la collaboration et communication dans l’équipe (les devs ne savent pas parler les gros bouffons mdrrr)
* Adopter des méthodes itératives et incrémentales pour déployer les changements liés à DevOps
* Mettre l’accent sur l’amélioration continue et la rétroaction en s’appuyant sur les principes du Lean et de l’agilité.
* Favoriser la culture de l’apprentissage et d’expérimentation en encourageant les équipes à tirer parti des échecs pour s’améliorer

**Roadmap de transition (template) :**

* Evaluer l’état actuel de l’organisation Audit
* Définir les objectif et les priorités de la transition
* Etablir un plan de formation et de dév des compétences de l’équipe (mise à niveau)
* Sélectionner et implémenter les outils et les processus adaptés aux besoins de l’organisation
* Mettre en place des métriques et des indicateurs de performance pour mesurer les progrès
* Planifier des revues régulières pour évaluer les progrès et ajuster la roadmap si nécessaire

**Résistance aux changements et adoptions des nouvelles pratiques**

* Identifier et comprendre les sources de résistance au changement
* Communiquer clairement les bénéfices et les avantages de la transformation DevOps
* S’assurer que les membres de l’équipe comprennent les nouvelles méthodes et les objectifs associées
* Fournir des formations et des ressources pour aider les employés à développer les compétences nécessaires
* Mettre en place des mécanismes de soutien et de feedback pour encourager l’engagement et l’adaptation.

**Enjeux techniques et sécurité**

* Sélectionner les outils appropriés pour répondre aux besoins spécifiques de l’organisation
* Standardiser les processus pour faciliter la collaboration et la communication entre les équipes
* Mettre en place une documentation claire et accessible pour aider les membres de l’équipe à comprendre les outils et les processus
* Encourager l’adoption d’outils et de processus automatisés pour réduire les erreurs et les inefficacités
* Assurer une intégration étroite entre les outils pour faciliter la gestion et le suivi des projets

**Maintenir la sécurité tout en accélérant le développement et le déploiement**

* Adopter l’approche « Security As Code » pour intégrer la sécurité dès le début du processus de développement
* Automatiser les tests de sécurité et les contrôles pour détecter rapidement les vulnérabilités
* Collaborer étroitement avec les équipes de sécurité pour comprendre les exigences et les contraintes
* Mettre en place des mécanismes de surveillance et d’alerte pour détecter et réagir rapidement aux incidents de sécurité
* Former les membres de l’équipe aux pratiques de développement sécurisé et aux risques associés

**Stratégie de gestion du changement**

**Etapes Manière Public**

Sensibilisation = présentations et ateliers = Toute l’équipe

Compréhension = Réunions d’information, Q&R = Membres concernés

Engagement = Discussions, retours d’expérience = Toute l’équipe

Consolidation = Célébrations, partage des succès = Toute l’équipe

**Formation et accompagnement des équipes tout au long du processus**

* Identifier les besoins en formations et en compétences pour chaque membre de l’équipe
* Proposer des formations adaptées aux rôles et aux responsabilités de chaque membre
* Mettre en place un programme de mentorat et de coaching pour soutenir l’apprentissage continu
* Encourager la collaboration et le partage des connaissances entre les équipes

**Utilisation d’indicateurs de performance pour suivre les progrès**

* Définir des KPI pertinents pour mesurer les progrès de la transformation DevOps
* Collecter et analyser régulièrement les données relatives aux KPI
* Partager les résultats avec l’ensemble de l’équipe pour favoriser la transparence
* Utiliser les données pour ajuster améliorer continuellement les processus DevOps
* Célébrer les améliorations et les succès pour encourager l’engagement de l’équipe