



Cours 3 : Les SIRH dans la stratégie d'entreprise et la gouvernance





Sommaire du cours

Les SIRH dans la stratégie d'entreprise et la gouvernance

Rôles et responsabilités des acteurs des SI

Management : Alignement sur la stratégie d'entreprise

Gouvernance des SI : Alignement stratégique

Alignement stratégique dans la gouvernance des SI

Gestion des risques des SI

Sécurité des systèmes d'information

Stockage des données et Data Manager

L'auditeur : Audit et évaluation des systèmes d'information

Gestion des données et de l'analytique RH

Suivi de performance et optimisation des SIRH

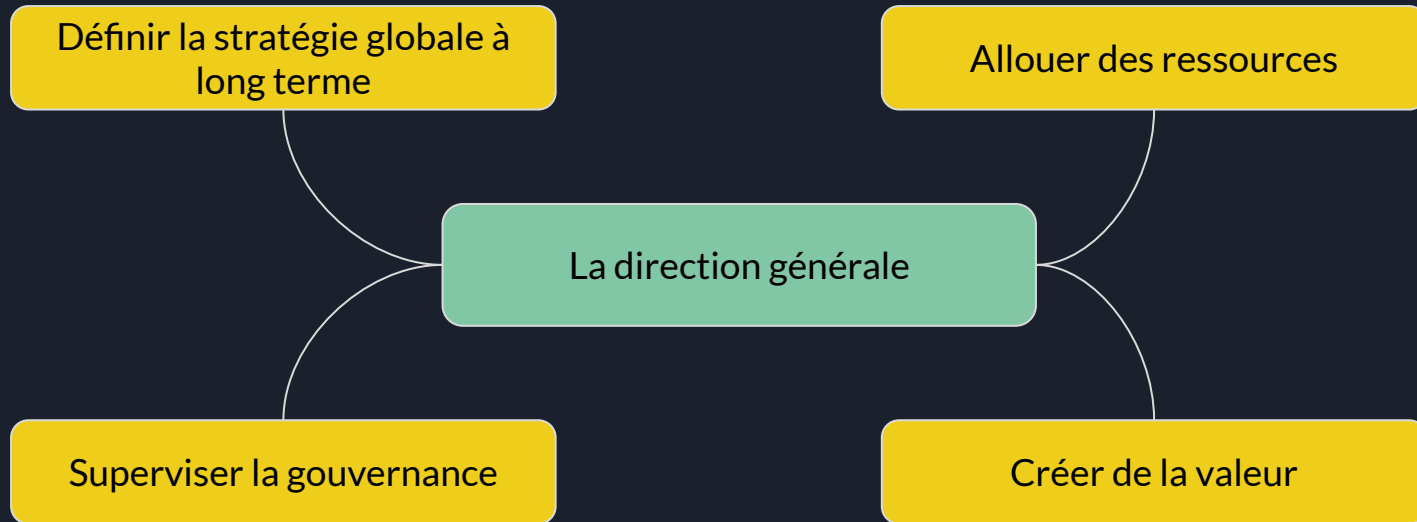
L'analyse de performance : exemple de KPIs

Méthodologie : DevOps appliqué au RH

L'amélioration continue

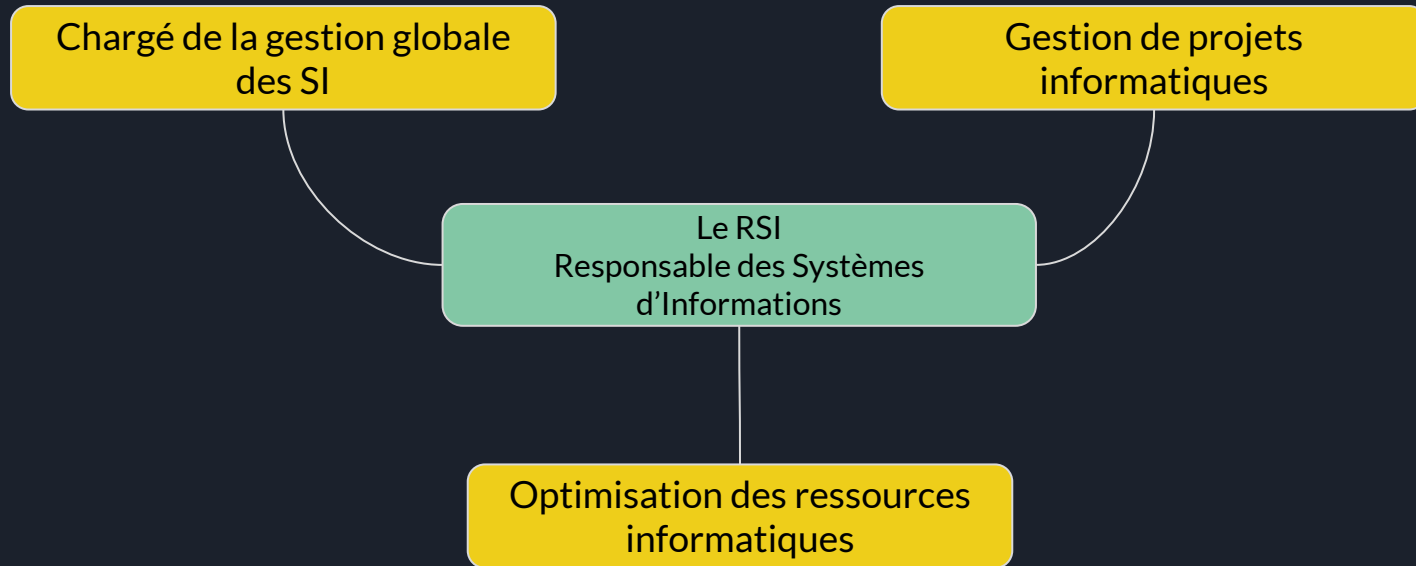


Rôles et responsabilités des SI



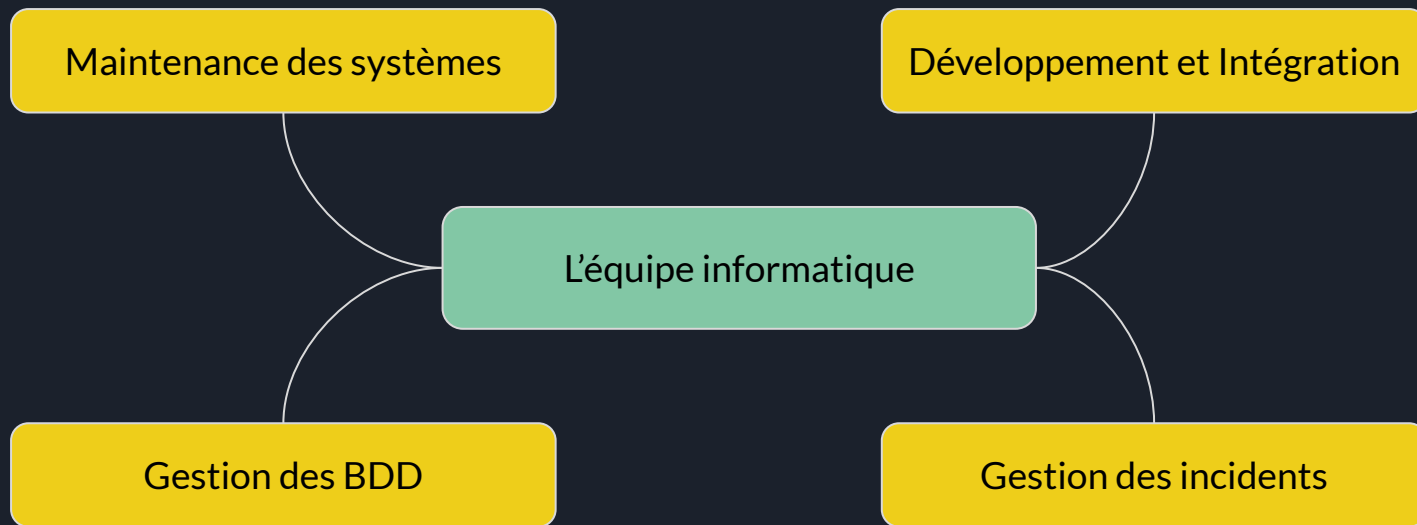


Rôles et responsabilités des SI



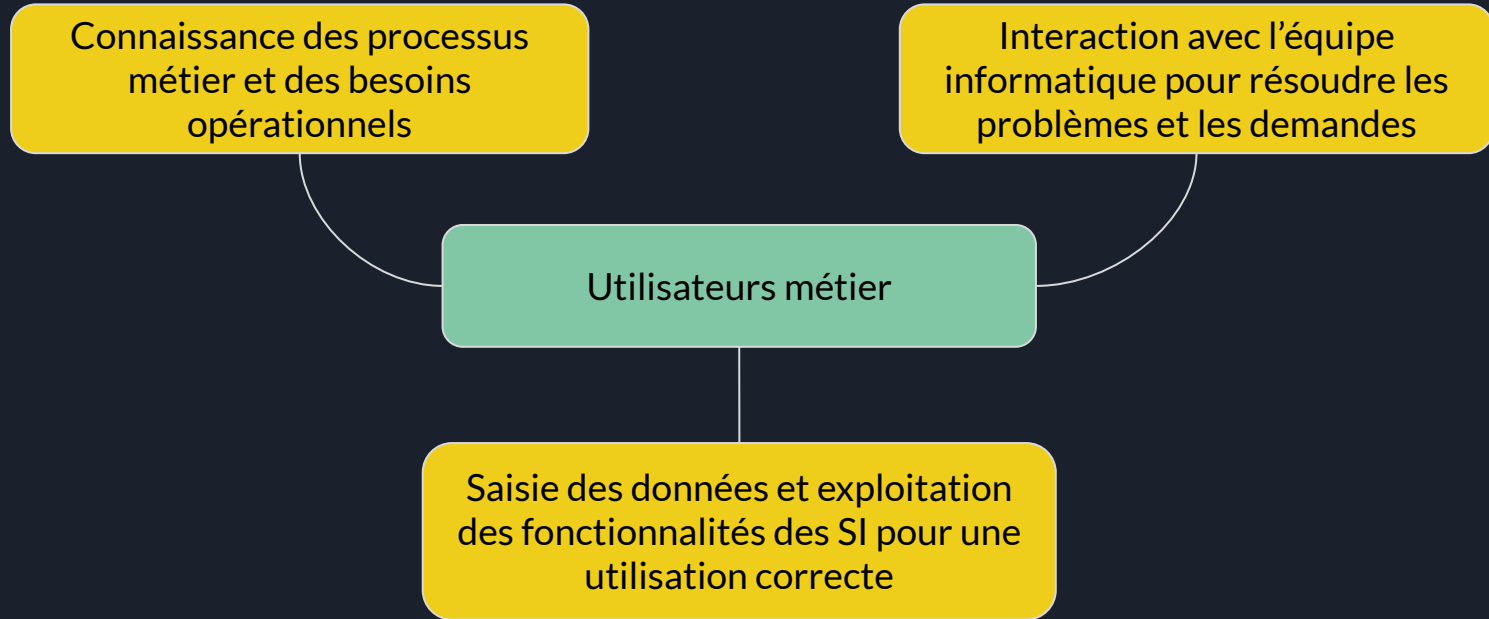


Rôles et responsabilités des SI



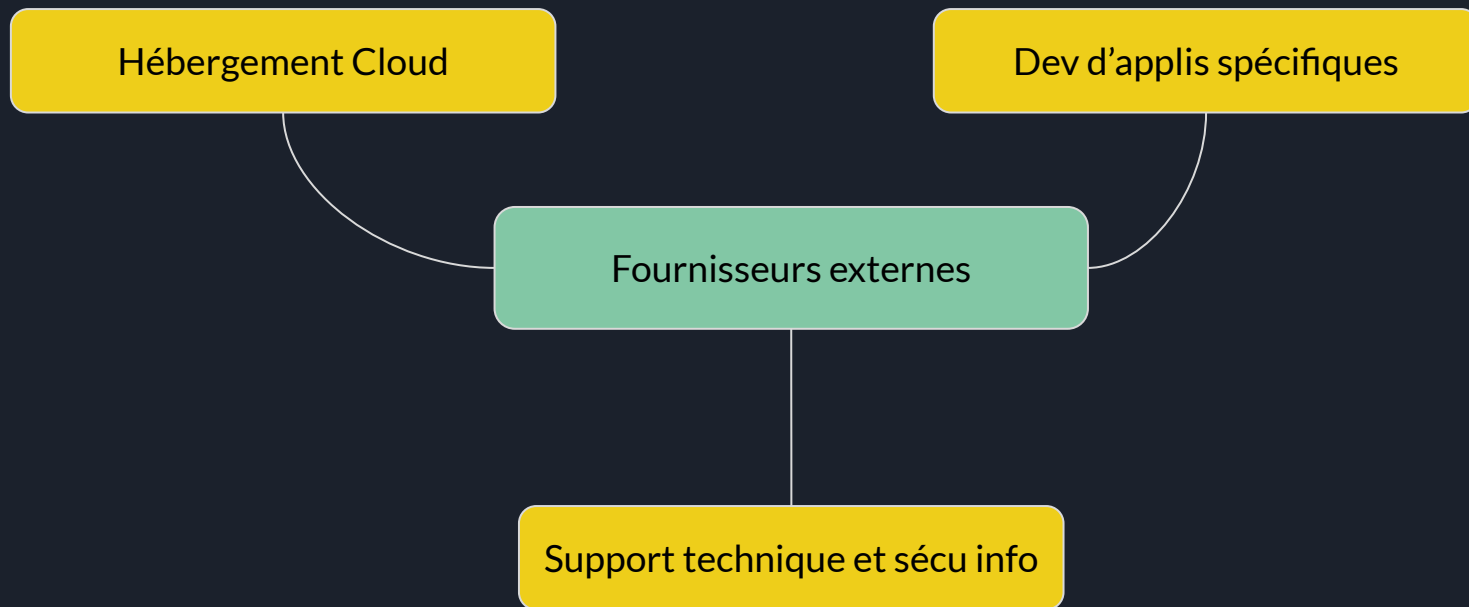


Rôles et responsabilités des SI






Rôles et responsabilités des SI





Etude de cas : Cartographie des rôles et responsabilités des SI dans vos entreprises

Au sein de vos entreprises, vous allez cartographier les différents services affiliés aux SI et SIRH.




Management : Alignement sur la stratégie d'entreprise

Les systèmes d'informations doivent être alignés sur la stratégie globale de l'entreprise pour maximiser leurs valeurs.

Identification des
objectifs stratégiques
de l'entreprise

Intégration des SI dans
les processus métiers

Communication/Collab
oration avec les parties
prenantes



Management : Alignement sur la stratégie d'entreprise

Identification des objectifs stratégiques de l'entreprise

Identifier les objectifs stratégiques afin de trouver comment les systèmes d'information peuvent les soutenir (Processus, Méthodes, Outils)

Déterminer les investissements nécessaires pour tendre vers les objectifs visés.



Management : Alignement sur la stratégie d'entreprise

Intégration des SI dans les processus métiers

Intégrer les SI dans les processus métiers pour les accompagner et faciliter leur travail.

Rationaliser les processus de travail et éviter les pertes de temps et la qualité des données.



Management : Alignement sur la stratégie d'entreprise

Communication/Collaboration avec les parties prenantes

Identifier plus rapidement les exigences des SI pour le besoin opérationnel

S'assurer que les SI respectent la conformité aux normes et réglementations.



Alignement stratégique dans la gouvernance des SI

Comprendre les besoins
de l'entreprise

La compréhension
claire des objectifs et
des besoins de
l'entreprise permet de
bien définir sa
roadmap et sa
stratégie.

Définitions des
contributions

La réalisation des
objectifs de
l'entreprise passe par
une gouvernance et
des contributions
clairement définies

Cohérence avec les
priorités

Pour garantir
l'alignement
stratégique, les
investissements
technologiques
doivent être
cohérents avec les
priorités de
l'entreprise.



Gestion des risques des SI

Identification des
risques

L'identification des
risques liés aux SI est
le premier pas pour
une gestion efficace
des SI

Evaluation des risques

L'évaluation des
risques informatiques
permet de
déterminer leur
gravité et de prioriser
les actions de gestion
des risques

Gestion des risques

Vise à réduire les
vulnérabilités des SI
et à garantir la
continuité des
opérations. Elle inclut
également des
mécanismes de
contrôle et de
surveillance pour
détecter et prévenir
les risques potentiels



Sécurité des systèmes d'information

Protection des données

Gestion des risques

Sécurité des réseaux

Gestion des incidents de
sécurité



Stockage des données et Data Manager

Le rôle de Data Manager est de garantir la disponibilité et l'intégrité des données de l'entreprise en gérant leur stockage, leur utilisation et leur processus. Les responsabilités du Data Manager comprennent :

- De s'assurer que les données sont stockées de manière sécurisée et conformes aux politiques de sécurité des données de l'entreprise.
- De déterminer les stratégies de sauvegarde et de récupération des données pour minimiser les pertes de données en cas de panne.
- De surveiller l'utilisation des données et garantir leur qualité pour assurer leur intégrité et leur utilisation
- D'assurer l'accès approprié aux données pour les utilisateurs autorisés tout en protégeant les données sensibles contre les accès non autorisés.



L'auditeur : Audit et évaluation des systèmes d'information

Identifier les forces et les faiblesses des SI

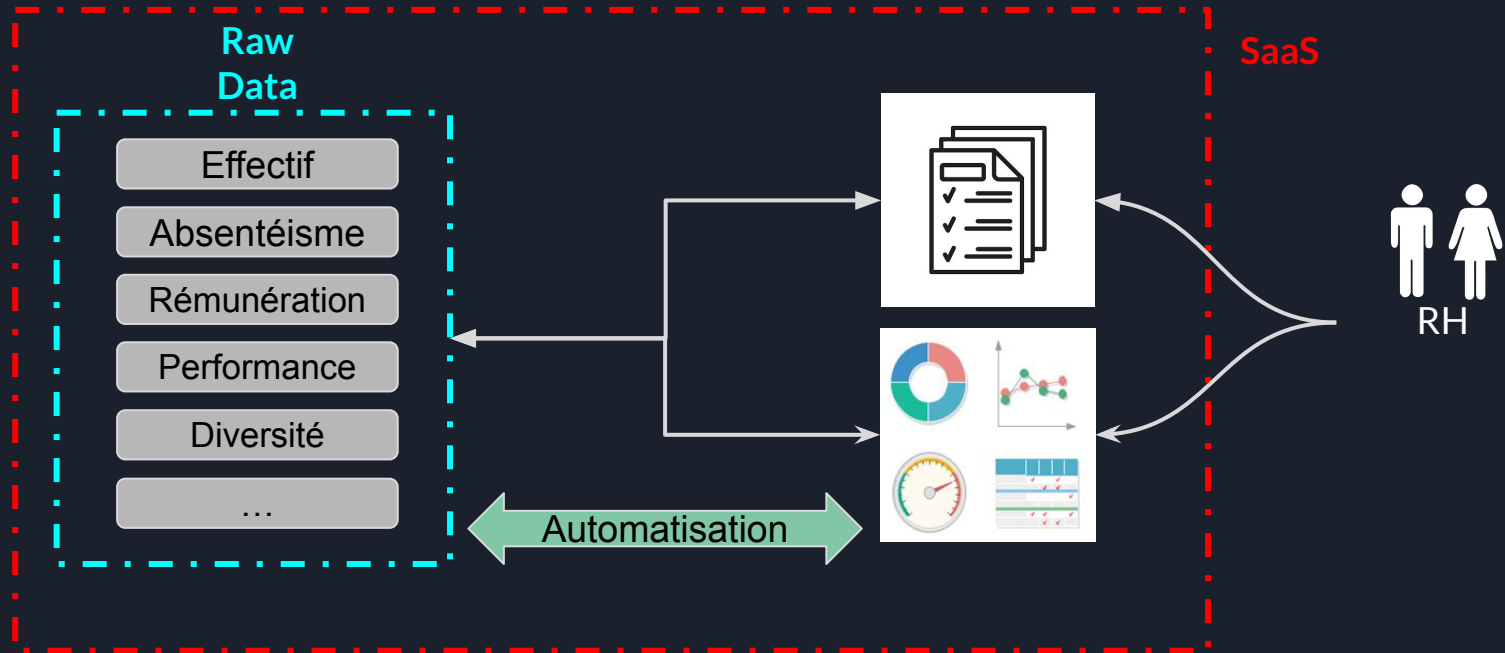
Evaluer la performance des systèmes d'information

Vérifier la conformité

Proposer des améliorations et des solutions

Gestion des données et de l'analytique RH

Reporting et génération de rapports



Tableaux de bord et visualisations

Gestion des données et de l'analytique RH

Analyse prédictive

Raw
Data

Effectif

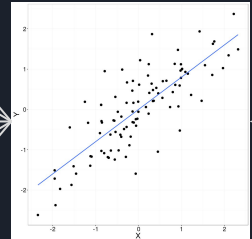
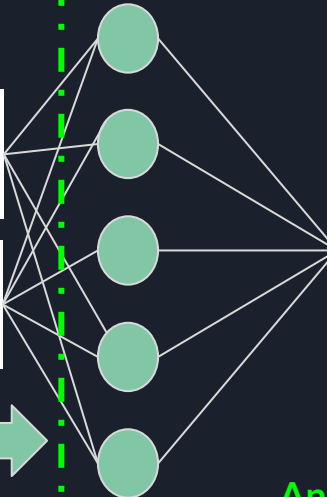
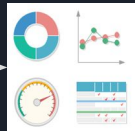
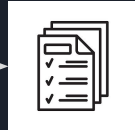
Absentéisme

Rémunération

Performance

Diversité

...



Automatisation

Analyse prédictive

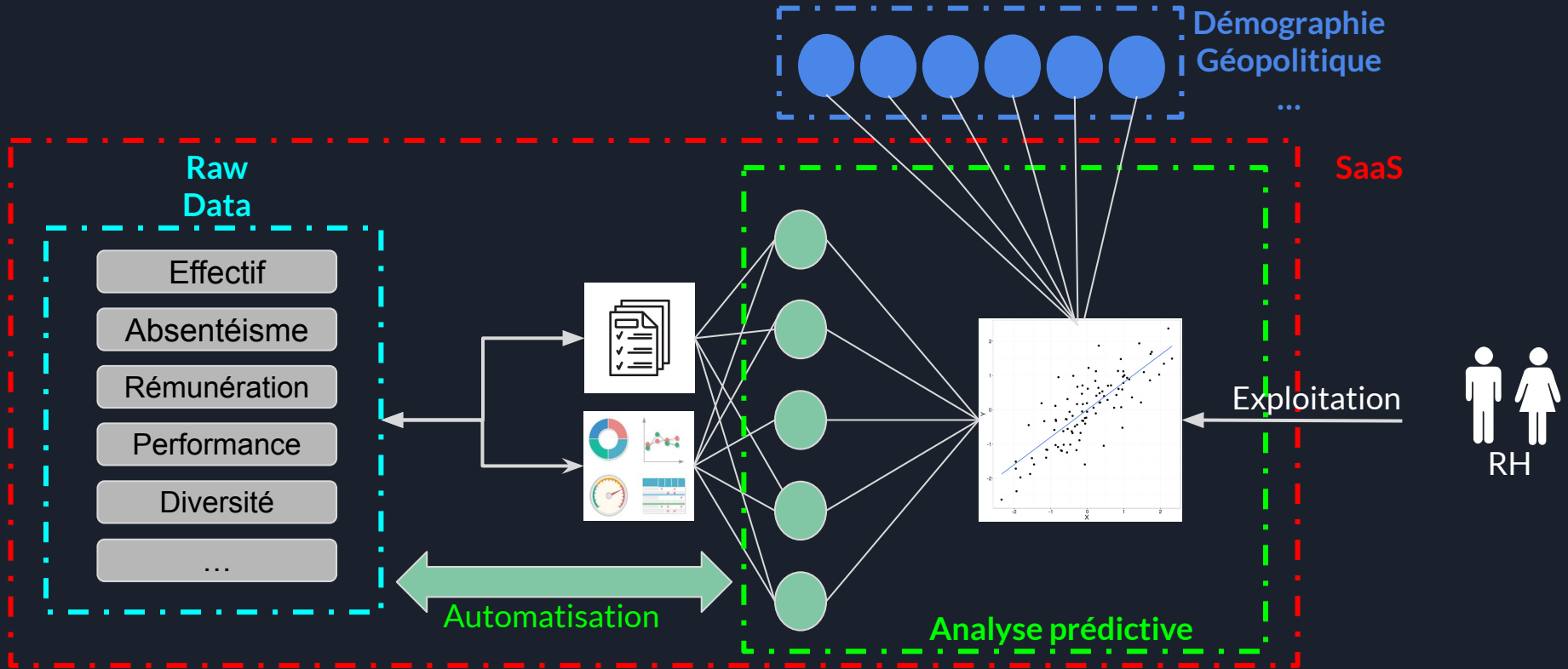
SaaS

Exploitation



Gestion des données et de l'analytique RH

Intégration de sources de données externes

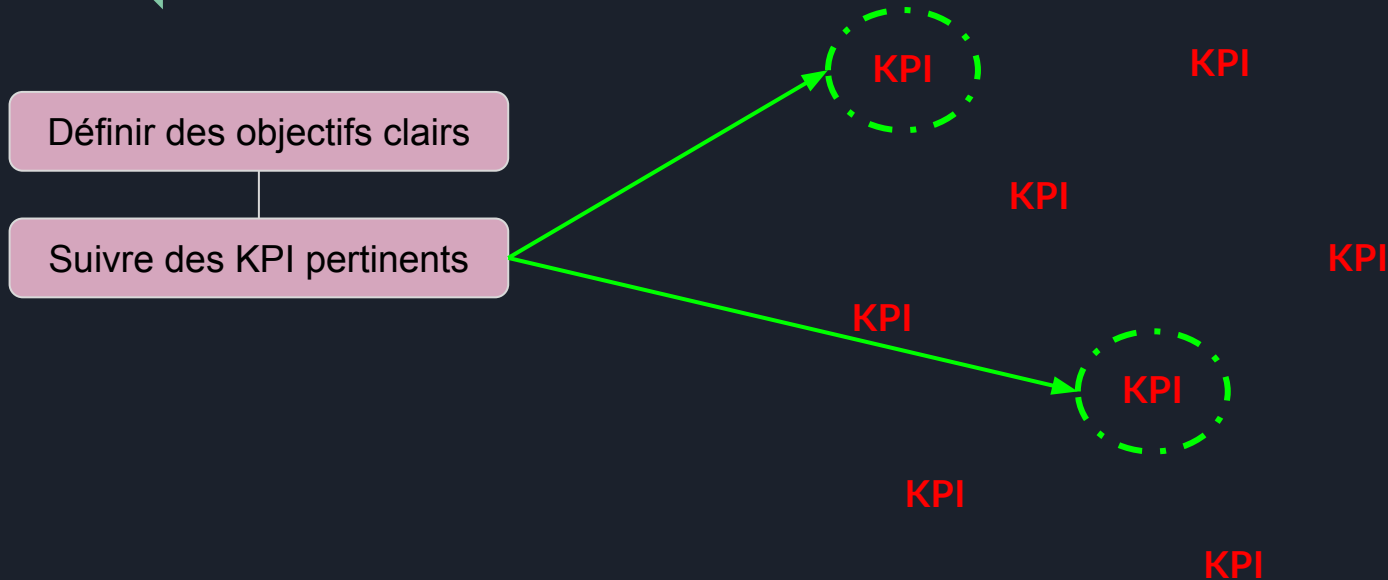





Suivi des performances et optimisation des SIRH

Définir des objectifs clairs

Suivi des performances et optimisation des SIRH



/!\ Certaines informations échappent à cette méthodologie /!\



L'analyse des performances : exemple de KPIs

Taux d'adoption du
SIRH

Efficacité
opérationnelle

Précision des
données

Taux de conformité

Taux de satisfaction
des employés

Réduction des coûts
administratifs

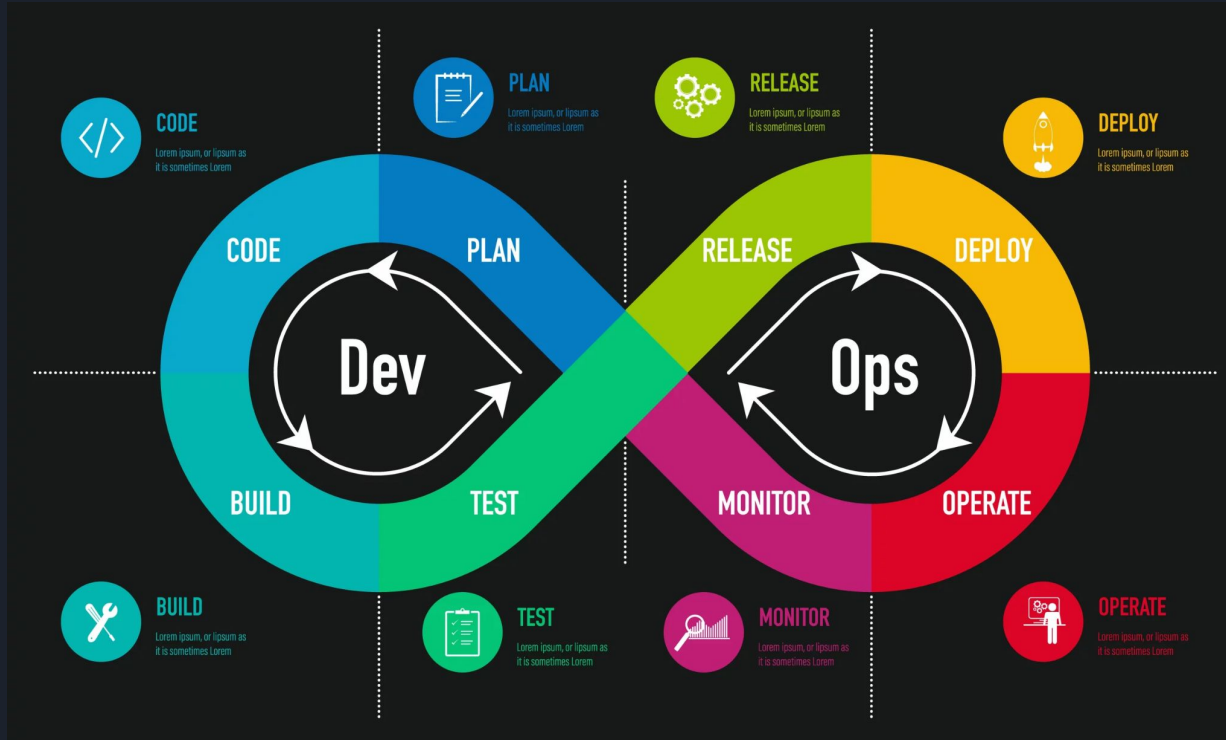
Taux
d'automatisation

Temps moyen de
résolution des
problèmes

Analyse de la
rétention des
talents

Taux d'utilisation
des fonctionnalités

Méthodologie DevOps appliquée au SI RH





L'intégration Continue (CI)

L'intégration continue (CI) est une pratique de développement de logiciel qui consiste à intégrer les modifications et les ajouts de code de manière régulière et automatisée dans le code source principal du projet. L'objectif de l'intégration continue est de s'assurer que le code source est stable et fonctionnel à tout moment, en détectant et en corrigeant les erreurs et les bugs de manière précoce.



L'intégration Continue (CI)

Pour mettre en place une intégration continue, il est nécessaire de configurer un environnement de développement et de déploiement automatisé, qui inclut des outils de build, de test et de déploiement. Ces outils permettent de construire, de tester et de déployer le code source de manière automatisée, en s'appuyant sur un système de contrôle de version (Git, SVN, etc.) et sur un serveur d'intégration continue (Jenkins, Travis CI, etc.).



L'intégration Continue (CI)

L'intégration continue s'appuie sur une stratégie de développement en continu, qui consiste à développer et à intégrer le code de manière continue et itérative, plutôt que de le faire de manière séquentielle et ponctuelle. Cette stratégie permet de réduire les temps de développement et de déploiement, de s'assurer de la qualité et de la stabilité du code, et de faciliter la collaboration et la communication au sein de l'équipe de développement.



Outils de build

Les outils de build sont des logiciels qui permettent de compiler, d'assembler et de packager le code source de l'application en vue de sa mise en production. Ils peuvent automatiser et optimiser différentes tâches de build, comme la génération de fichiers binaires, la création de fichiers de configuration, l'inclusion de dépendances, la minification de code, etc. Exemples d'outils de build : Maven, Gradle, Ant, Webpack.

Remarque :

Il existe deux grandes catégories de langages de programmation : les langages compilés et les langages interprétés.

Langages compilés vs Langages interprétés

Langages compilés

Doivent être
compilés avant
exécution

Exécution plus
rapide

Exécution plus
efficace

Compilation
longue et
obligatoire

Sécurité du code

Environnement
de dev lourd



Langages interprétés

Plus simple
d'utilisation

Pas de
compilation

Exécution plus
lente
(interpréteur)

Modification
flexible

Code lisible par
tous

Environnement
de dev léger

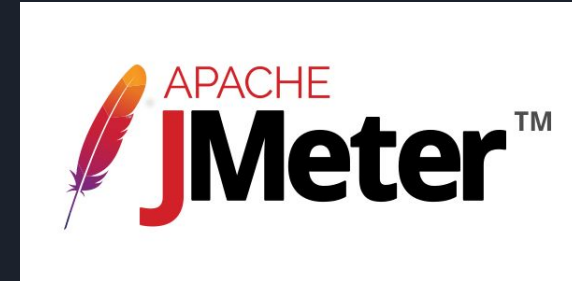




Outils de test

Les outils de test sont des logiciels qui permettent de vérifier et de valider le bon fonctionnement et la qualité de l'application. Ils peuvent exécuter différents types de tests, comme les tests unitaires, les tests d'intégration, les tests de performance, les tests de sécurité, etc. Les résultats des tests sont généralement reportés dans des rapports, qui permettent de suivre l'état de l'application et de détecter les éventuels problèmes.

JUnit Test**NG**





Tests de performance

Les tests de performance sont des tests qui mesurent les indicateurs de performance d'un système, tels que sa vitesse, sa capacité de traitement, sa fiabilité, etc. Ils permettent de s'assurer que le système est capable de gérer les charges de travail prévues et de répondre aux exigences de performance définies.

Les tests de performance sont souvent exécutés par les testeurs ou les administrateurs système au cours de la phase de déploiement, pour valider les performances du système en conditions réelles.



Tests de sécurité

Les tests de sécurité sont des tests qui vérifient la sécurité d'un système, en détectant et en corrigeant les failles et les vulnérabilités qui pourraient être exploitées par des attaquants. Ils permettent de s'assurer que le système est protégé contre les menaces externes et internes, et qu'il respecte les règles de sécurité en vigueur. Les tests de sécurité sont souvent exécutés par des experts en sécurité ou des sociétés de testing spécialisées, pour valider la sécurité du système avant son déploiement.



Tests d'acceptation

Les tests d'acceptation sont des tests qui vérifient que le système répond aux besoins et aux attentes des utilisateurs ou des clients. Ils permettent de s'assurer que le système est facile à utiliser, ergonomique et fonctionnel, et qu'il respecte les standards et les normes en vigueur. Les tests d'acceptation sont souvent exécutés par les utilisateurs ou les clients finaux au cours de la phase de déploiement, pour valider l'adéquation du système aux besoins et aux attentes de l'organisation.



Déploiement Continu (CD)

Le déploiement continu (Continuous Deployment en anglais) est une pratique de développement logiciel qui consiste à automatiser et à intégrer de manière continue les étapes de build, de test et de déploiement d'une application. Le but du déploiement continu est de permettre une livraison rapide et fréquente de nouvelles fonctionnalités et de corrections de bugs, tout en maintenant un niveau élevé de qualité et de stabilité du code.



Déploiement Continu (CD)

Pour mettre en œuvre le déploiement continu, il est nécessaire de mettre en place une chaîne d'intégration et de déploiement (CI/CD en anglais) qui permet de gérer de manière automatisée et transparente les différentes étapes du processus de déploiement. La CI/CD peut être basée sur des outils de build, de test et de déploiement tels que Jenkins, Travis CI, CircleCI, etc.

CD sur un serveur de test

CD dans le cloud



Déploiement continu sur un serveur de staging

Dans ce scénario, chaque fois qu'un développeur valide un commit sur la branche de développement, le processus de CI/CD déclenche automatiquement la compilation et les tests de l'application, puis déploie le code sur un serveur de staging (environnement de préproduction). Les tests de non-régression et les validations manuelles peuvent être effectuées sur ce serveur avant de déclencher le déploiement sur le serveur de production.



Déploiement sur un serveur de production

Dans ce scénario, le processus de CI/CD déploie automatiquement le code sur le serveur de production dès qu'il est validé sur la branche de développement. Cette approche est plus risquée, car elle nécessite de mettre en place des tests de qualité et de stabilité très rigoureux pour éviter les erreurs de déploiement.



Déploiement continu sur le cloud

Dans ce scénario, le processus de CI/CD déploie automatiquement l'application sur un environnement de cloud computing, comme AWS ou Google Cloud. Cette approche permet de bénéficier d'une scalabilité et d'une flexibilité importantes, ainsi que d'une facilité de gestion et de maintenance.



Outils de déploiement

Les outils de déploiement sont des logiciels qui permettent de déployer l'application sur un ou plusieurs serveurs de production. Ils peuvent automatiser et optimiser différentes tâches de déploiement, comme le transfert de fichiers, la configuration de l'application, le redémarrage de services, la mise à jour de la base de données, etc. Ils peuvent également gérer les différentes versions de l'application et les rollbacks en cas de problème. Exemples d'outils de déploiement : Jenkins, Github Action, Gitlab CI, ArgoCD ...