

Quels sont les enjeux de l'intégration de la sécurité au sein du modèle DevOps ?



3DEXPERIENCE®



Université Paris – Saclay
Site d'Evry Val d'Essonne

Référent :
Eric LAHARGOUE

Dassault Systèmes
Vélizy – Villacoublay

Maître d'apprentissage :
Jérôme DUFFET

FAIZAN MOUHAMAD
M2 MIAGE Alternance
2020 – 2021

1. Epigraphe

« *Le DevOps n'est pas une fin en soi, mais un processus infini d'amélioration continue* »

■ Jez Humble

« *Les premiers pas vers une industrialisation des développements consistent généralement en la mise en place d'une intégration continue. Alors que celle-ci est souvent vue comme un aboutissement, elle n'est qu'une première étape pour parvenir à des réalisations efficaces et maîtrisées.* »

■ Farhdine Boutzakhti

Table des matières

1.	Epigraphe	2
2.	Remerciements.....	5
3.	Glossaire & liste des abréviations	6
4.	Liste des figures.....	10
5.	Liste des annexes	11
6.	La fiche de bilan et de synthèse	12
6.1.	Présentation de l'activité en entreprise	12
6.1.1.	L'entreprise d'accueil	12
6.1.2.	Le maître d'apprentissage.....	14
6.1.3.	Résumé des travaux proposés par l'entreprise	14
6.1.4.	Les travaux effectués en entreprise	15
6.2.	Présentation et synthèse du sujet de mémoire de l'apprenti	16
6.2.1.	Présentation du (ou des) sujet de mémoire sur lesquels les apports sont novateurs	
	16	
6.2.2.	Ce qui est déjà connu sur les sujets traités dans le mémoire	17
6.2.3.	Ce que le travail de mémoire apporte de nouveau.....	17
6.2.4.	Utilisation potentielle des travaux de votre sujet de mémoire	17
6.2.5.	Principales perspectives des travaux	18
7.	Introduction	19
8.	Le DevOps.....	20
8.1.	Qu'est-ce que l'approche DevOps ?	20
8.2.	Les enjeux / caractéristiques de cette méthode de développement	22
8.3.	Avantages / Inconvénients / Limites	28
9.	DevOps + Sécurité = DevSecOps.....	31
9.1.	Qu'est-ce que le DevSecOps ?	31
9.2.	L'importance d'intégrer la sécurité au cœur du DevOps	32
9.3.	Avantages / Inconvénients.....	34
10.	Mise en place d'un modèle de développement DevSecOps	39
10.1.	Plan d'action de mise en place du DevSecOps.....	39
10.1.1.	Intégration de la sécurité dans la boucle DevOps.....	39
10.1.2.	Etapes à mettre en œuvre pour instaurer cette nouvelle méthodologie	42
10.1.3.	Autres points à prendre en considération	44
10.2.	Outils recommandés et nécessaires	45
11.	Application au sein de l'équipe DevOps de Dassault Systèmes	48

11.1.	Chiffrement des communications SSL	48
11.2.	Mise en place de Remote Apps	49
12.	Conclusion	50
13.	Bibliographie	51
14.	Annexes	55

2. Remerciements

Ce mémoire de fin d'études est le résultat d'un travail de recherche de plusieurs mois. En préambule, je souhaite adresser mes remerciements aux personnes qui m'ont apporté leur soutien et qui ont contribué à l'élaboration de ce mémoire de recherche, travail indispensable pour la validation de mon master.

J'adresse mes remerciements tout particulièrement à mon maître d'apprentissage, **M. Jérôme DUFFET**, qui a su prendre le temps pour me guider dans mon travail, pour m'avoir fait découvrir de nouveaux métiers et de nombreux types de missions, et qui m'a permis de progresser et de monter en compétence aussi bien d'un point de vue technique qu'humain.

Je remercie notamment mes collègues de bureau, pour leur disponibilité, leur expérience et leur partage de connaissances, dans un cadre agréable et néanmoins studieux. J'adresse également mes remerciements à toutes les personnes que j'ai pu rencontrer tout au long de l'année et qui m'ont fait découvrir la diversité des métiers au sein de Dassault Systèmes.

Je tiens à fortement remercier fortement **M. Eric LAHARGOUE**, mon tuteur enseignant et **Mme Judith BENZAKKI**, pour leur disponibilité et leur aide dans la réalisation de ce mémoire. Je tenais aussi à remercier mon chargé de mission, **M. Xavier CRENN** ainsi que l'ensemble de l'équipe pédagogique de l'université d'Evry Val d'Essonne.

Enfin, je remercie ma famille, mon entourage proche et mes amis pour leur soutien et leurs encouragements ainsi que pour leur implication et aide dans la relecture.

3. Glossaire & liste des abréviations

A

- ANSSI** : Agence Nationale de la Sécurité des Systèmes d'Information
- API** : Application Programming Interface, ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications
- Artefact** : Information créée, produite, modifiée ou utilisée par un homme dans la mise au point d'un système informatique
- AWS** : Amazon Web Services, division du groupe Amazon spécialisée dans les services de cloud computing à la demande pour les entreprises et particuliers

B

- Bogue (bug)** : Défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement

C

- CI** : Continuous Integration, pratique de développement logiciel dans laquelle les développeurs fusionnent fréquemment les nouvelles modifications de code dans la branche de code principale.
- CIS** : Center for Internet Security, organisme à but non lucratif qui exploite la puissance d'une communauté informatique mondiale pour protéger les organisations publiques et privés contre les cybermenaces
- CD** : Continuous Delivery, déploiement fréquent et automatisé de nouvelles versions d'applications dans un environnement de production.
- Cloud** : Informatique en nuage, accès à des services informatiques via Internet à partir d'un fournisseur (Amazon, Google, Microsoft)
- Covid-19** : Maladie infectieuse émergente de type virale causée par la souche de coronavirus
- Cyberattaque:** Acte malveillant envers un dispositif informatique via un réseau cyberspace

D

- DAST** : Dynamic Application Security Testing, méthode de sécurité qui permet de détecter les vulnérabilités et les faiblesses de sécurité d'une application en cours d'exécution

DevOps : Ensembles de pratiques pour automatiser les processus entre développeurs et exploitants

DevSecOps : Amélioration de l'approche DevOps avec la prise en compte de l'aspect sécurité au sein du cycle de vie de l'application

DS : Dassault Systèmes

DSI : Directeur des Systèmes d'Information

F

Framework : Ensemble d'outils et de composants logiciels, organisés conformément à un plan d'architecture et des patterns, à la base d'un logiciel ou d'une application

I

IAST : Interactive Application Security Testing, méthode de sécurité qui surveille une application en cours à l'aide d'instruments afin de détecter d'éventuels failles ou vulnérabilités

IDE : Integrated Development Environment, ensemble d'outils de développement qui permet d'augmenter la productivité des programmeurs qui développent des logiciels

IAAS : Infrastructure As A Code, modèle de cloud computing destiné aux entreprises où le fournisseur cloud gère le matériel serveur, les couches de virtualisation, le stockage et les réseaux

ISO : Organisation Internationale de Normalisation, ensemble de règles définies et approuvées par un organisme reconnu, garantissant un niveau d'ordre optimal dans un contexte donné

IT : Information Technology, domaine technique du traitement de l'information

L

Linting : Pratique visant à améliorer la qualité du code d'un langage de programmation en particulier et ainsi améliorer la maintenabilité de celui-ci

K

- Kanban** : Méthode de gestion des connaissances relatives au travail, qui met l'accent sur une organisation de type « juste-à-temps » en fournissant l'information ponctuellement aux membres de l'équipe afin de ne pas les surcharger
- Kubernetes** : Système open-source qui vise à fournir une plateforme permettant d'automatiser le déploiement, la montée en charge et la mise en œuvre de conteneurs d'application sur des clusters de serveurs

M

- Malware** : Programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté
- MFA** : Multi-Factor Authentication, méthode d'authentification qui exige de l'utilisateur qu'il fournisse plusieurs facteurs de vérifications (au moins deux) pour accéder à une ressource
- Monitoring** : Surveillance active d'un système informatique et de l'état du réseau informatique à des fins préventives, dans le but de détecter des failles ou des dysfonctionnements

P

- Phishing** : Technique utilisée par des fraudeurs pour obtenir des informations personnelles dans le but de perpétrer une usurpation d'identité
- PLM** : Product Lifecycle Management, solution pour la gestion du cycle de vie d'un produit
- POC** : Proof Of Concept, réalisation ayant pour vocation de montrer la faisabilité d'un procédé ou d'une innovation

R

- R&D** : Recherche & Développement
- RASP** : Run-time Application Security Testing, outil de sécurité branché sur une application afin de la protéger de toutes failles de sécurité
- RDS** : Remote Desktop Services, architecture centralisée qui permet à un utilisateur de se connecter sur un ordinateur distant afin d'accéder à une application ou à des ressources spécifiques
- Release** : Version spécifique d'un code logiciel
- RGPD** : Règlement Général sur la Protection des Données

S

- SAST** : Static Application Security Testing, méthode de sécurité qui permet de trouver des failles de sécurité dans le code source des applications plus tôt dans le cycle de vie du développement logiciel
- SCA** : Software Composition Analysis, outil conçu pour analyser les composants open source en détectant les licences logicielles, les dépendances dépréciées, les vulnérabilités connues et les exploits potentiels dans une base de code
- SI** : Système d'Information
- SQL** : Structured Query Language, langage informatique normalisé servant à exploiter des bases de données relationnelles
- SSH** : Secure Shell, protocole réseau cryptographique permettant d'exploiter des services réseau en toute sécurité sur un réseau non sécurisé
- SSO** : Single Sign-On, méthode d'authentification permettant à un utilisateur d'accéder à plusieurs applications informatiques en ne procédant qu'à une seule authentification

T

- TLS** : Transport Layer Security, protocole de sécurisation des échanges par réseau informatique
- TTM** : Time To Market, délai de mise sur le marché correspondant au temps qu'il faut entre la conception d'un produit et sa mise en vente

W

- WAF** : Web Application Firewall, type de pare-feu qui protège le serveur d'applications Web dans le backend contre diverses attaques

4. Liste des figures

Figure 1: DASSAULT SYSTEMES, campus de Vélizy	12
Figure 2: Organigramme de DS	13
Figure 3: Qu'est-ce que DevOps ?	20
Figure 4: DevOps, le mur de la confusion	21
Figure 5: DevOps, boucle infinie	22
Figure 6: Les outils DevOps par étape	24
Figure 7: les différents types de tests.....	25
Figure 8: Concepts de CI / CD	26
Figure 9: Tableau de bord d'un outil de monitoring, Solarwinds	26
Figure 10: Evolution de la sécurité au sein de DevOps	30
Figure 11: Différence entre DevOps et DevSecOps	31
Figure 12: Evolution de la cybercriminalité dans le monde.....	32
Figure 13: DevOps + Automation = Security	34
Figure 14: Sanctions du RGPD	35
Figure 15: Les 3 piliers de la sécurité.....	36
Figure 16: Freins à la démarche DevSecOps.....	37
Figure 17: Projet d'investissement DevSecOps.....	38
Figure 18: Exemple de linting de code JS	39
Figure 19: Fonctionnement d'un WAF	41
Figure 20: Pratiques de sécurité au sein de la boucle DevOps	42
Figure 21: Principe de fonctionnement du reverse proxy.....	48

5. Liste des annexes

Annexe 1: Infographie DevOps	55
Annexe 2: Enquête MicroFocus	56

6. La fiche de bilan et de synthèse

6.1. Présentation de l'activité en entreprise

6.1.1. L'entreprise d'accueil

Dassault Systèmes est une entreprise d'éditeur de logiciels spécialisée dans la conception 3D, le maquettisme numérique 3D et les solutions pour la gestion du cycle de vie d'un produit (PLM).

Aujourd'hui, elle est devenu le leader mondial sur le marché des solutions logicielles de gestion de vie des produits (PLM) en offrant aux entreprises et aux particuliers les univers virtuels nécessaires à la conception d'innovations durables. **Ses solutions transforment pour ses clients, la conception, la fabrication et la maintenance de leurs produits.** [1]

Les solutions collaboratives de Dassault Systèmes permettent de promouvoir l'innovation sociale et offrent de nouvelles possibilités d'améliorer le monde réel grâce aux univers virtuels.

Avec des ventes dans plus de 140 pays, le Groupe apporte de la valeur à plus de **190 000 entreprises de toutes tailles dans toutes les industries** et s'affirme sur de nombreux secteurs d'activités. DS peut compter parmi ses partenaires en France : **AIRBUS, RENAULT** ou encore **BOEING**, qui a signé en 2017 un contrat d'une valeur d'un milliard de dollars d'une durée de 30 ans, renouvelable tous les 10 ans.

Le siège social de Dassault Systèmes se trouve sur le campus de Paris à Vélizy-Villacoublay sous la direction de **Bernard CHARLES**, qui est aussi Vice-Président du Conseil d'Administration. Il s'agit d'un campus dynamique et propice à l'innovation. On y trouve six bâtiments qui sont identifiés par les principaux éléments « Eau », « Terre », « Air », « Feu », puis « Métal » et « Terre Europa » symbolisant la vie, la force, l'inspiration et l'innovation.



Figure 1: DASSAULT SYSTEMES, campus de Vélizy

L'entreprise obtient un chiffre d'affaires de **4,451 milliards d'euros** pour l'année 2020 correspondant à 9% de hausse par rapport à l'année précédente et compte **21 560 salariés** à son actif. DS fait son entrée au sein de l'indice CAC 40 à partir du 24 septembre 2018, les actions de l'entreprise sont cotées à la Bourse de Paris, concrétisant ainsi la forte croissance de l'activité de DS depuis sa création.

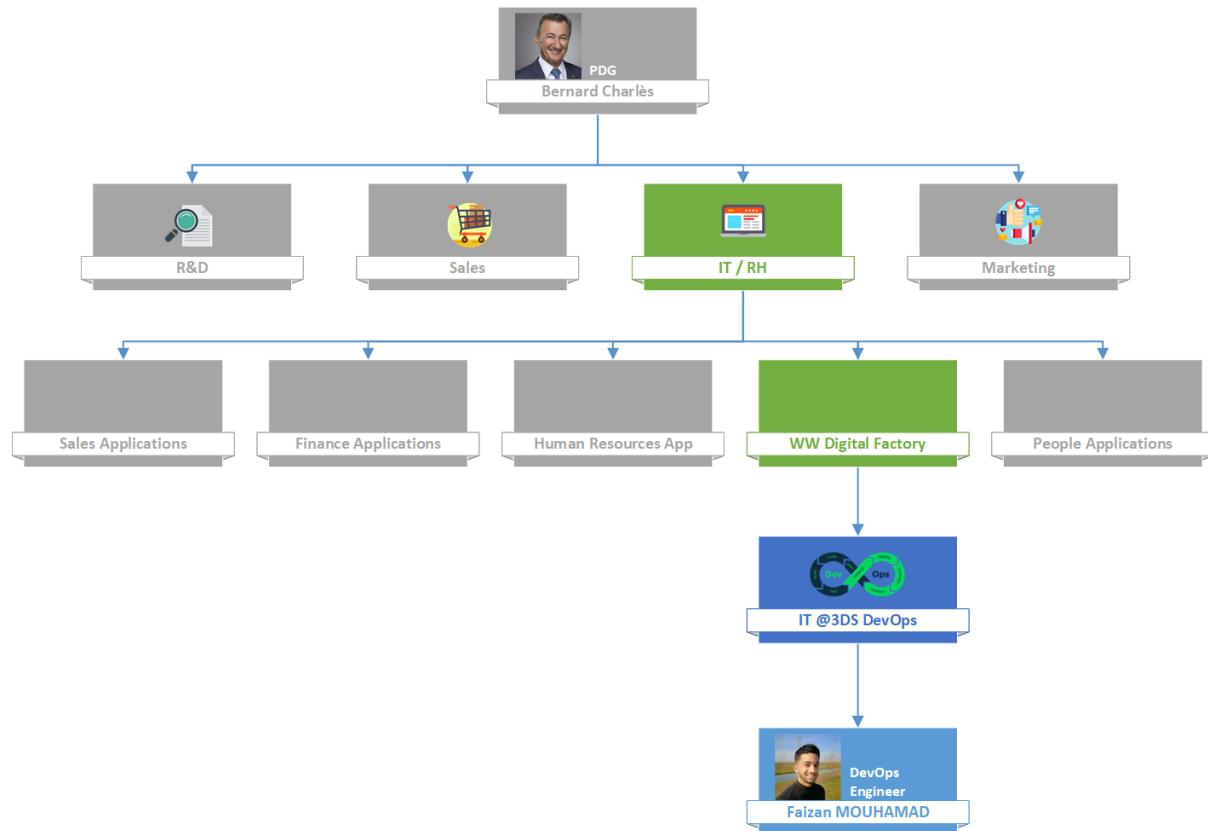


Figure 2: Organigramme de DS

L'équipe dans laquelle je travaille est l'« IT @3DS DevOps » dont les missions qui lui sont affectées sont les suivants :

- Veiller au bon fonctionnement des application IT, People, Finance, RH déployés en interne ;
- Déployer et livrer des applications et de nouvelles fonctionnalités rapidement ;
- Développer et apporter de l'innovation aux produits existants ;
- Assurer la sécurité du parc informatique.

6.1.2. Le maître d'apprentissage

Mon maître d'apprentissage, M. Jérôme DUFFET, est le manager de l'équipe IT @3DS DevOps, ses missions principales sont les suivantes :

- Faire le suivi hebdomadaire des différents projets et applications que l'on gère ;
- Définir les spécifications fonctionnelles en fonction des besoins qui lui sont remontés ;
- Mettre en place des processus de livraison de code performants et les documenter ;
- Faire le point avec les différents membres de son équipe, situés en France mais également en Inde ;
- Assurer le support et la haute-disponibilité des applications en production.

6.1.3. Résumé des travaux proposés par l'entreprise

Lors de mon apprentissage chez Dassault Systèmes, les principaux travaux qui m'ont été proposés sont les suivants :

- **Automatisation de l'installation** d'une solution d'indexation et de recherche d'information, Exalead CloudView ;

Auparavant, l'ensemble des installations se faisait manuellement par des membres de notre équipe située sur le site de Pune en Inde et durait en moyenne une vingtaine de minutes. Cependant, avec la forte augmentation du nombre de demandes, il n'était plus possible de poursuivre ce mode d'installation.

- **Mise en place d'un processus de déploiement automatique** de code pour les applications se basant sur CloudView ;

Les équipes qui travaillent sur CloudView, développent du code et des plugins afin d'apporter des fonctionnalités supplémentaires à l'outil. Ces derniers, qui sont créés dans leur environnement de développement doivent ensuite être intégrés dans les environnements d'intégration, de qualification et de pré-production afin d'être validés et testés puis mis en production.

Ce déploiement se faisait manuellement par la copie des différents fichiers d'un serveur à l'autre et d'autres actions manuelles qui étaient plus ou moins longues. La conséquence de ce mode de déploiement est que parfois, on avait des serveurs qui n'avaient pas tous le même niveau de code ou qui étaient différents les uns des autres.

- **Amélioration de la sécurité** des solutions / logiciels gérés par mon équipe.

Pour l'ensemble des solutions que nous déployons à l'aide de nos outils, notre manager nous a demandé de prendre en compte l'aspect sécurité et d'améliorer le code afin d'être protégé face aux vulnérabilités. Par exemple, pour CloudView, l'authentification se faisait par défaut avec le compte « admin » créé lors de l'installation et dont le mot de passe est le même sur l'ensemble des serveurs, ou encore des connexions n'étaient pas chiffrées et étaient donc vulnérables à des attaques du type Man in The Middle (l'attaque de l'homme du milieu).

6.1.4. Les travaux effectués en entreprise

Lors de mes deux premières années d'apprentissage chez DS, j'ai dû me former sur les outils DevOps utilisés au sein de notre équipe (Git, Jenkins, SaltStack) et également monter en compétence dans le développement en Python.

Afin de réaliser l'automatisation de l'installation de CloudView, j'ai travaillé avec de nombreuses équipes dont la plupart sont situées sur le site de Pune en Inde, afin de bien comprendre et d'analyser les besoins et les spécificités de chacun.

J'ai ensuite développé des scripts qui automatisent l'ensemble des tâches qui étaient réalisées à la main, c'est-à-dire l'installation des binaires, la création des répertoires avec les bons droits et l'application de règles de sécurité afin de restreindre l'accès à la solution.

L'année suivante, j'ai dû mettre en place un processus de déploiement automatique de code dans CloudView, qui n'existe pas auparavant. Pour cela, j'ai eu la chance de travailler dans les locaux d'Exalead qui se trouve à La Madeleine à Paris et avoir l'aide de l'équipe R&D afin de parfaire la solution que je devais mettre en place.

Enfin, cette année, qui est ma dernière année d'apprentissage, j'ai eu comme mission d'améliorer la sécurité de CloudView. Tout d'abord, j'ai mis en place un reverse-proxy sur les serveurs CloudView, pour d'une part chiffrer l'ensemble des communications via le protocole TLS mais d'autre part restreindre l'accès aux services avec des règles de restriction au niveau du « firewall » (pare-feu).

J'ai également mis en place l'authentification par Active-Directory, qui permet de se connecter à CloudView en utilisant son compte Windows de DS. Cela permet de ne plus utiliser le compte « admin » mais également de pouvoir tracer l'ensemble des interventions réalisées par un utilisateur. Le mot de passe par défaut a été remis à jour et stocké dans une solution de gestion de mot de passe afin d'être utilisé en cas d'urgence uniquement.

Aujourd'hui, toutes ces solutions sont en production et sont utilisées quotidiennement par de nombreuses équipes.

De nombreux enseignements du master ont pu me servir dans la réalisation de mes missions, notamment la matière *Sécurité des Systèmes d'Information* qui m'a permis de monter en compétences sur l'analyse de risques et la mise en place de bonnes pratiques ISO 27002 au sein de mon entreprise ou encore *Architecture des Systèmes d'Information* qui m'a permis d'améliorer ma vision du système d'information et de mieux comprendre le fonctionnement global du système ainsi que comment l'améliorer.

6.2. Présentation et synthèse du sujet de mémoire de l'apprenti

6.2.1. Présentation du (ou des) sujet de mémoire sur lesquels les apports sont novateurs

Problématique : Quels sont les enjeux de l'intégration de la sécurité au sein du modèle DevOps ?

Le lundi 11 janvier 2021, le directeur général de l'ANSSI, Guillaume Poupard, indique que le nombre de cyberattaques visant les entreprises ou des institutions a considérablement explosé en France voire multiplié par 4. [2] En effet, on constate que 91% des organisations françaises ont été la cible d'au moins une cyberattaque en 2020, marqué par la crise sanitaire exceptionnelle de la Covid-19.

Face à cette tendance inquiétante, les entreprises n'ont d'autre choix que de réagir et d'améliorer la protection de leurs systèmes face aux vulnérabilités qu'elles présentent. [3]

De plus, avec l'évolution constante de l'informatique et de la technologie, les entreprises cherchent à être proactive, c'est-à-dire de livrer le plus de nouvelles fonctionnalités de manière stable le plus rapidement possible afin de faire le maximum de profits.

L'approche DevOps est en phase avec cette philosophie. En effet, en éliminant les barrières entre les équipes de développement (Dev) et de l'exploitation des systèmes (Ops), on offre d'une part de la rapidité dans le cycle de développement des applications grâce aux notions d'intégration et de déploiement continu. D'autre part, on assure la stabilité et la fiabilité des systèmes en réduisant les interventions humaines grâce à l'automatisation.

Cependant, à travers l'application de cette approche, on se rend compte que la prise en compte de la composante sécurité indépendamment du cycle de vie de nos applications ralentit le processus de développement et ne répond pas aux exigences en termes de vitesse de développement et de sécurité de l'application.

C'est pourquoi, à travers ce mémoire de fin d'études, nous allons analyser les enjeux de l'intégration de la sécurité au cœur de l'approche DevOps. Cette étude permettra aux entreprises qui font déjà du DevOps d'être convaincu de l'importance de la prise en compte de cette composante au sein du cycle de vie de l'application mais également de connaître les différents moyens de la mettre en place. Et pour les autres sociétés, elle permettra d'avoir des arguments de taille et des exemples en faveur de la mise en place de cette méthode de développement ainsi que de ses impacts.

Dans le monde de l'informatique et surtout avec le contexte sanitaire actuel, les sujets comme celui-ci prennent de plus en plus de place du fait de l'importance de la sécurité au sein des systèmes d'informations afin de se protéger des nombreuses attaques informatiques. Les entreprises cherchent à solidifier leurs systèmes afin de garantir la fiabilité et la haute disponibilité de leurs solutions, notamment à Dassault Systèmes. En effet, l'une des priorités majeures de l'année 2020 dans mon entreprise était d'améliorer la sécurité de nos systèmes d'informations pour se protéger des cyberattaques qui ne cessent d'explorer.

Ce sujet rentre pleinement dans mes activités en entreprise de cette dernière année d'apprentissage. En effet, dans le département où je suis, nous travaillons sur l'amélioration de la sécurité des composants et logiciels que nous déployons notamment en sécurisant davantage les étapes d'intégration et de déploiement continue de code.

6.2.2. Ce qui est déjà connu sur les sujets traités dans le mémoire

Il existe de nombreux articles sur le DevSecOps ou encore le DevOps, de l'explication du principe de ses deux approches à la mise en place par quelques entreprises. Ces références sont très complètes et détaillées par des experts et pour la plupart, d'actualité. Cependant, on ne trouve aucun mémoire qui ne traite de ces deux sujets afin de démontrer l'importance de l'intégration de la composante sécurité au sein du modèle DevOps.

Afin de construire un mémoire complet et de qualité, la principale source de documentation que j'utiliserais seront des articles réalisés par des entreprises avec une grande renommée et spécialisée dans l'IT.

6.2.3. Ce que le travail de mémoire apporte de nouveau

Ce mémoire de fin d'études aura pour finalité de convaincre les entreprises que la composante sécurité est primordiale et qu'elle doit être intégrer dans le cycle de développement des applications, afin d'assurer aussi bien la rapidité et la stabilité des déploiements mais aussi garantir la sécurité de la solution. En effet, vu la hausse exponentielle des cyberattaques en France, l'application de ces bonnes pratiques de sécurité permettront peut-être à des entreprises de se protéger de ces attaques mais également de parfaire leurs activités.

L'étude de ce sujet permettrait d'apporter des arguments en faveur de l'intégration de la sécurité au sein du cycle de développement de nos applications mais également d'analyser les enjeux et les coûts liés à cette approche.

6.2.4. Utilisation potentielle des travaux de votre sujet de mémoire

Ce travail de recherche peut servir à court terme aux entreprises informatiques qui ont déjà un mode de développement DevOps en place afin de la faire évoluer vers une approche plus sécurisée, mais aussi globalement à l'ensemble des entreprises qui souhaiteraient solidifier la sécurité de leurs systèmes. Ces méthodes s'appliquent aux entreprises utilisant l'agilité avec la méthode DevOps. Les entreprises qui ont opté pour une autre méthode de développement pourront, elles aussi, s'en servir afin de sortir de l'effet tunnel et avoir une méthode de développement plus complète, rapide et sécurisée.

L'ensemble de ces travaux sont utilisables par mon entreprise d'accueil, Dassault Systèmes, puisque ceux-ci sont expérimentés actuellement dans le cadre de projet afin de démontrer leurs avantages sur le SI et qu'on a différentes équipes qui appliquent actuellement l'approche DevOps comme méthode de développement d'applications.

6.2.5. Principales perspectives des travaux

La sécurité informatique est un sujet d'actualité qui touche plusieurs entreprises. Avec la croissance accrue du nombre de cyberattaques, les entreprises ont tout intérêt à solidifier la protection de leurs SI afin de garantir la disponibilité de leurs activités. Au vu de l'évolution des nouvelles technologies, il est certain que de nouvelles approches meilleures que le DevOps ou le DevSecOps naîtront, que ce soit dans un futur proche, ou même lointain. L'analyse qui est faite dans mon mémoire évoque les moyens et technologies existants, je pense cependant qu'il existera d'autres technologies au futur, qui feront oublier celles actuelles.

7. Introduction

Avec l'évolution constante de l'informatique et de la technologie, les entreprises cherchent à développer et à fournir des logiciels de qualité de manière plus efficace, plus rapide et plus rentable pour les clients.

D'après un rapport établi par le cabinet informatique CollabNet VersionOne, 97% des entreprises informatiques utiliseraient des méthodes de développement agiles afin d'améliorer la proactivité de l'entreprise. [4] Cette dernière met en œuvre des pratiques spécifiques dans l'ensemble de son organisation, notamment en améliorant sa stratégie, son système d'innovation et d'organisation ou encore son personnel et sa culture. Le but final étant de constituer une organisation capable de s'adapter rapidement à des changements inattendus de son environnement, en conservant une continuité stratégique, opérationnelle et humaine.

Parmi les différentes méthodes agiles existantes, la méthode DevOps est la démarche la plus à même à répondre aux besoins cités ci-dessus. Cette dernière est une démarche qui améliore la relation entre les équipes de développements (Dev) ainsi que les exploitants (Ops) et qui dote toutes ses équipes d'un ensemble d'outil qui permettent l'automatisation du déploiement de code avec un niveau de qualité paramétrable.

D'autre part, on constate également que les attaques informatiques ne cessent d'augmenter, notamment depuis janvier 2020 avec la pandémie du coronavirus. En effet, la cybercriminalité a augmenté de 30 000% en 2020, passant de 1200 à plus de 380 000. Ce sont principalement de l'hameçonnage (*phishing*) et des logiciels malveillants (*malwares*), des sites malicieux qui ciblent des utilisateurs à distance. [5]

C'est pourquoi, il est primordial pour les entreprises de nos jours de se préparer à d'éventuelles attaques et donc d'améliorer la protection de leurs systèmes pour éviter la catastrophe. Cependant, l'intégration de la composante sécurité ne doit pas perturber la productivité de l'entreprise en engendrant des ralentissements du cycle de développement ou encore de la sécurité des applications.

A travers ce mémoire, nous essaierons de trouver des solutions afin de permettre à une entreprise de conserver un niveau de productivité élevé tout en intégrant la composante sécurité au sein du cycle de développement de ses applications. C'est pourquoi je vais répondre à la problématique suivante :

Quels sont les enjeux de l'intégration de la sécurité au sein du modèle DevOps ?

Je commencerais par présenter la méthode DevOps en étudiant ses enjeux, caractéristiques ainsi que ses limites. Puis, je continuerais ce mémoire en me focalisant sur la méthode DevSecOps notamment en analysant les différents avantages et inconvénients que présentent cette approche. J'expliquerai ensuite comment mettre en place une approche DevSecOps, c'est-à-dire comment mettre en place un cycle de développement rapide, efficace et de qualité et qui intègre également la sécurité au cœur de ce processus. Enfin, nous enrichirons ce mémoire en s'appuyant sur des exemples et projets mis en place dans le cadre de mon apprentissage chez Dassault Systèmes.

8. Le DevOps

8.1. Qu'est-ce que l'approche DevOps ?

Le terme **DevOps** (**D**evelopment & **O**perations) est un mouvement informatique qui privilégie la **collaboration** entre les équipes de développement (dev) et d'exploitation (ops) pour toute solution IT, en **automatisant le processus de livraison de logiciels et les changements d'infrastructure**. Cet ensemble de pratiques a pour objectif de pouvoir créer, tester, publier des logiciels plus rapidement et de manière plus fiable.

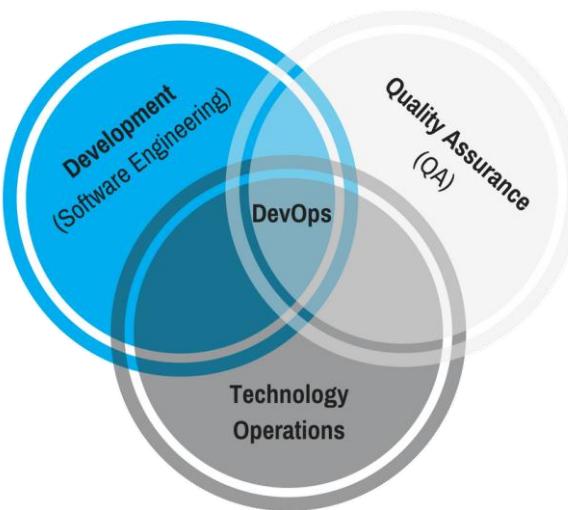


Figure 3: Qu'est-ce que DevOps ?

Lorsque l'on parle de « Dev », on fait référence aux équipes impliquées dans la fabrication du logiciel avant qu'il ne soit déployé en production, c'est-à-dire les développeurs, les testeurs, les Product Owners et enfin les équipes de tests. Quant aux « Ops », il s'agit de l'ensemble des personnes impliquées dans l'exploitation et la maintenance de la production, c'est-à-dire les ingénieurs systèmes, réseaux, les administrateurs des bases de données ou encore le personnel de sécurité.

Cette relation plus étroite entre Dev et Ops se reflète dans chaque phase du cycle de vie DevOps : planification logicielle initiale, codage, développement, test, publication, déploiement, opérations et surveillance continue. Elle génère de façon constante des retours clients, ce qui renforce le potentiel d'amélioration lors du développement, des tests et du déploiement. La publication accélérée et permanente des modifications ou ajouts de fonctionnalités en est un exemple. [6]

L'approche DevOps vise à créer une culture et un environnement dans lesquels la conception, les tests et la diffusion de logiciels peuvent être réalisées rapidement, fréquemment et efficacement. DevOps n'est pas seulement une méthodologie, c'est une véritable philosophie de travail.

Comment l'approche DevOps est-elle née ?

A l'origine, les exigences posées pour un logiciel étaient clairement définies à l'avance. La définition du produit même était également stable. D'un côté, nous avons les développeurs qui sont chargés de produire de l'innovation et délivrer les nouvelles fonctionnalités aux utilisateurs dès que possible. De l'autre côté, on a les équipes opérationnelles qui sont chargés de s'assurer que les utilisateurs ont accès à un système stable, rapide et réactif. Bien que le but ultime de ces deux pôles est de rendre l'utilisateur satisfait des systèmes qu'ils fournissent, leurs visions sur la façon de le faire restent diamétralement opposées. [7]

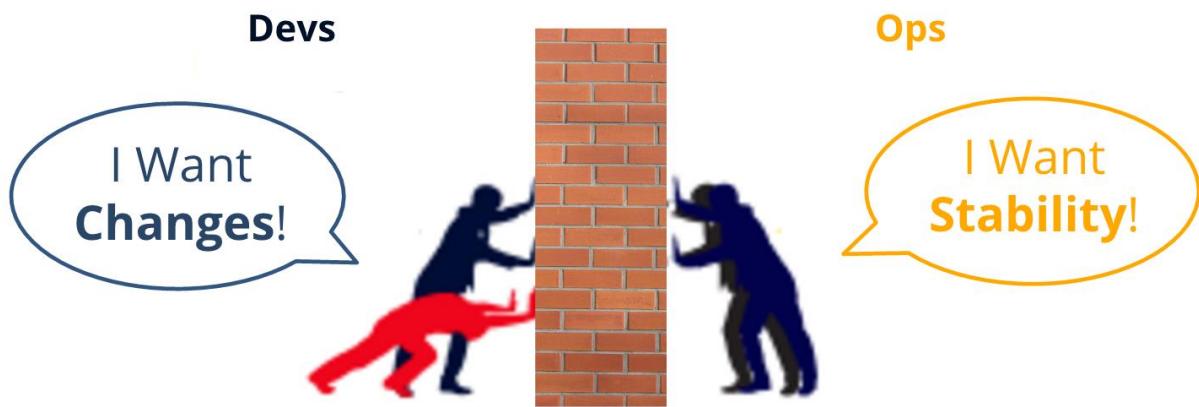


Figure 4: DevOps, le mur de la confusion

Auparavant, cela ne posait pas problème que les Dev et Ops travaillaient de leur côté de façon isolé. En effet, les deux travaillaient en collaboration uniquement pendant les rares phases de déploiement en production. Les Dev étaient conscients que cette livraison en production était leur seul moment de délivrer l'ensemble de leurs nouvelles fonctionnalités, au risque de devoir attendre le prochain déploiement. Et du côté des Ops, ils avaient suffisamment de temps pour tester les nouvelles fonctionnalités et préparer sereinement la prochaine livraison en production.

Cependant, avec la forte évolution des contraintes business, les utilisateurs finaux veulent des nouvelles fonctionnalités très rapidement. Les développeurs ne peuvent plus attendre des créneaux précis pour délivrer le produit comme auparavant. Et de l'autre côté, les équipes d'exploitants doivent être prêt plus rapidement à déployer tout le travail des développeurs mais ils doivent également garantir la stabilité des systèmes en production et la qualité de ce qui est délivrée.

L'approche DevOps permet donc, à travers un ensemble de pratiques, de mettre fin à la barrière entre les équipes de Dev et Ops et de parvenir à l'équilibre entre **l'innovation** et la **stabilité**.

8.2. Les enjeux / caractéristiques de cette méthode de développement

Toutes les directions informatiques ont comme préoccupation de satisfaire toujours mieux et toujours plus vite leurs directions métiers. L'objectif pour eux est d'améliorer leurs modèles de fonctionnement et leurs organisations afin de devenir plus rapides, flexibles et sans cesse continuer à améliorer le « Time to Market ». [8]

En s'appuyant sur sa capacité à industrialiser et automatiser les tâches et à limiter les opérations manuelles, la pratique DevOps a pour objectif de réduire au maximum les délais et les coûts de mise à disposition des infrastructures, de déploiement des applications et de prendre en compte de nouvelles fonctionnalités en production.

Dans la partie suivante, on va s'intéresser plus en détails sur la méthode de développement mise en place et appliquée au quotidien par les ingénieurs DevOps.

Quels sont les étapes dans un workflow de développement de type DevOps ?

Le DevOps est essentiellement un ensemble de pratiques qui sont effectués en boucle par les ingénieurs, de la conception de l'application jusqu'à son déploiement et son exploitation.

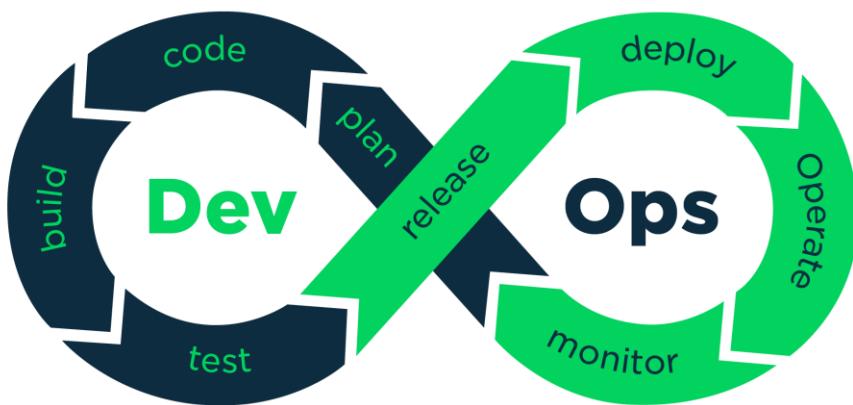


Figure 5: DevOps, boucle infinie

Parmi les ensembles de pratiques visibles sur la figure ci-dessus, nous retrouvons :

- **Plan**, phase de planification du projet, mise en place du cahier des charges :
 - Définition de la valeur commerciale et des exigences ;
 - Planification des indicateurs, des exigences, mesures commerciales, plan, timing ou encore de la politique de sécurité ;
 - Jira et Git peuvent être utilisés pour le suivi des problèmes connus et la gestion des projets.

- **Code**, phase de développement en suivant les spécifications définies :
 - Conception logicielle et création du code logiciel ;
 - Utilisation d'un langage de programmation, d'un IDE tel que Microsoft Visual Studio Code ou encore des outils de gestion de version de code tels que Github, Gitlab ou encore Bitbucket.
- **Build**, création d'une version exécutable du code :
 - Compilation et intégration du code en vue de sa mise en production ;
 - Création d'un « package » requise pour la livraison du produit à l'aide d'outils tels que Docker, Ansible, SaltStack ou encore Maven.
- **Test**, validation des fonctionnalités :
 - Ensemble d'opérations pouvant être manuels ou automatisées ;
 - Test unitaire, test de non-régression, test d'acceptation, test de performance, de charge...
 - Assurer une qualité du code optimale à l'aide de logiciels tels que JUnit, Selenium ou encore Vagrant.
- **Release**, création d'une version du produit testé et validé :
 - Empaqueter l'ensemble des travaux testés et validés en vue d'un déploiement ;
 - Configuration de la livraison ;
 - Utilisation de gestionnaires universels de paquets comme Artifactory ou encore Nexus.
- **Deploy**, phase de déploiement de la release
 - Livraison de la release en production ou dans un environnement ciblé ;
 - Déploiement et promotion des applications ;
 - Cette phase peut inclure des outils de gestion, de planification ou encore d'automatisation de la mise en production des produits tels que Puppet, Jenkins, Kubernetes ou encore Docker.
- **Operate**, contrôler le fonctionnement du produit déployé et de l'application :
 - Gestion des logiciels déployés en production ;
 - Activités de configuration et de préparation de l'infrastructure ;
 - Garantie de l'uniformité entre les différents serveurs ;
 - Solutions pour répondre à ces différents points sont nombreuses tels que Ansible, Chef, Puppet ou encore SaltStack.
- **Monitor**, surveiller l'état de fonctionnement de l'application et des métriques définies :
 - Collecte de métriques permettant la surveillance et l'analyse de logiciels et/ou de serveurs ;
 - Analyse et amélioration des performances de l'infrastructure ;
 - Identification d'un problème précis d'une livraison particulière et analyse de l'impact sur l'utilisateur final ;
 - Outils de supervision : Nagios, Shinken ou encore Grafana.

Dans la figure ci-dessous, on retrouve des exemples d'outils qui peuvent être mis en place et utilisés par les ingénieurs dans un workflow de développement de type DevOps :

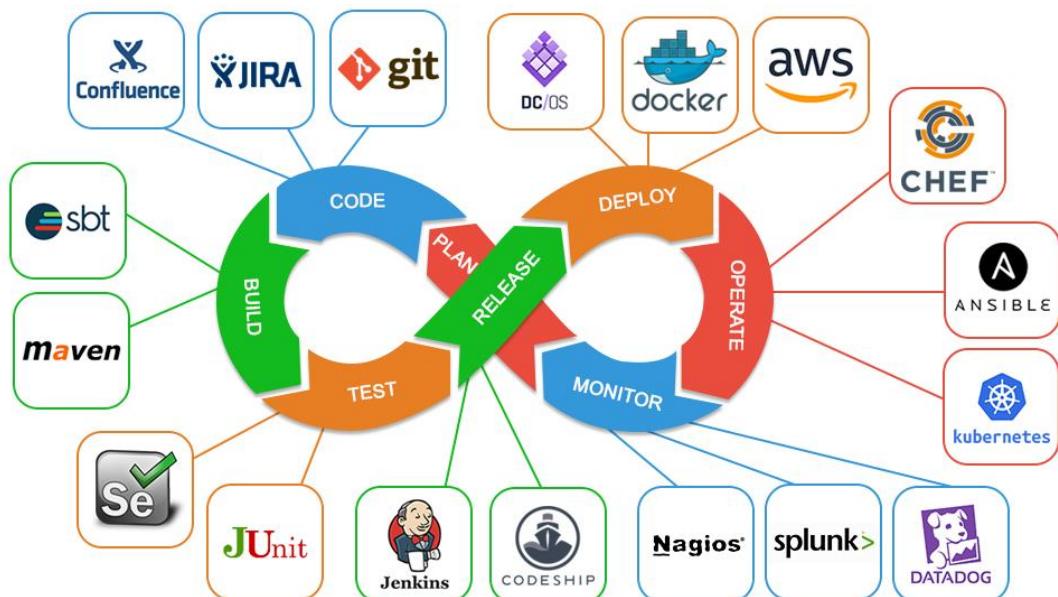


Figure 6: Les outils DevOps par étape

L'ensemble de ces pratiques forme le cycle de vie DevOps et sont appliquées en continu afin d'améliorer la vitesse de déploiement d'une nouvelle fonctionnalité en production tout en garantissant la qualité de ce qui est livrée. De ce fait, une entreprise qui met en place une approche DevOps de manière efficace peut être capable de livrer en continu de nouvelles fonctionnalités pour ces logiciels sans pour autant douter de la stabilité de l'application.

Concepts DevOps

A travers les nombreuses étapes constituant le cycle de vie DevOps, plusieurs concepts sont nés et portent sur une ou plusieurs phases du cycle de développement :

Tests continus automatisés : mettre en place une série de scénario de tests automatiques qui seront planifiés lors de l'ajout ou la modification du code de l'application. Cette méthode permet de rapidement identifier les erreurs afin de pouvoir les fixer et garantir une qualité de code élevée. Il existe de nombreux types de tests qui peuvent être mis en place comme par exemple les tests de non-régression (garantir qu'un changement dans le code n'entraîne aucune régression dans l'application) ou encore de charge (assurer que l'application fonctionne de la même manière avec 10 ou 100 utilisateurs).

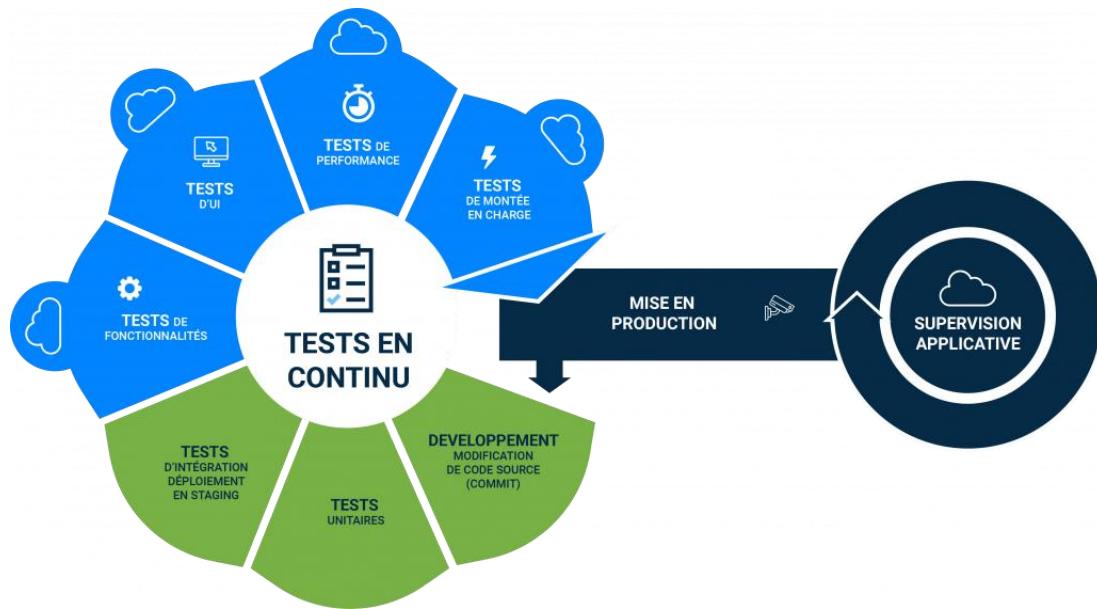


Figure 7: les différents types de tests

Intégration continue (CI) : les développeurs intègrent régulièrement les modifications qui sont apportées au code. A chaque modification du code, l'application est recréée automatiquement et différents niveaux de tests sont appliqués afin de garantir que le changement apporté ne crée pas de régression. En complément avec les tests continus automatisés, l'intégration continue permet de trouver et corriger les bugs plus rapidement, améliorer la qualité du code délivré et également de réduire le temps nécessaire à la validation de nouvelles mises à jour. A travers ce concept, on est donc capable d'assurer la stabilité du code de la branche principale et donc de notre application.

Livraison continue (CD) : cette phase consiste à automatiser l'empaquetage du code contenant les nouvelles modifications, préalablement testées et validées, et les publier dans un référentiel tel que Github ou encore des registres de conteneurs tels que Docker dans le but de pouvoir être déployé dans un environnement de production. Le processus de livraison continue permet de résoudre les problèmes de visibilité et de communication entre l'équipe de développement et l'équipe métier. Son objectif est donc de simplifier au maximum le déploiement du nouveau code. [9]

Déploiement continu : en supplément à l'intégration et à la livraison continue, la phase de déploiement continu gère le transfert automatique des modifications des développeurs depuis le référentiel contenant les releases jusqu'à l'environnement de production. Ce processus permet notamment de soulager les équipes d'exploitation surchargées par de nombreuses tâches manuelles qui ralentissent la distribution des applications.

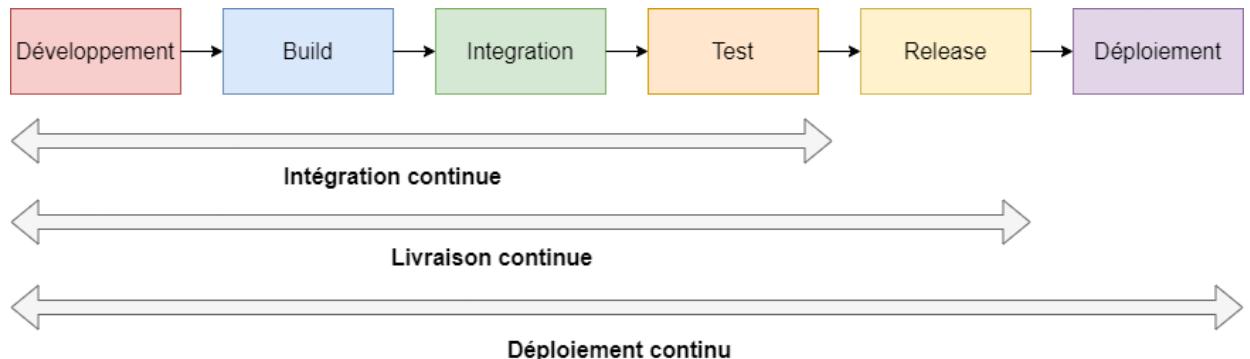


Figure 8: Concepts de CI / CD

Surveillance continue : connu également sous le nom de « monitoring », ce concept implique de superviser l'ensemble du processus de développement depuis la planification, le développement, l'intégration et les tests, le déploiement et les opérations. Le monitoring permet d'avoir une vue complète et en temps réel de l'état des applications, des services et de l'infrastructure dans l'environnement de production. L'objectif du monitoring est de permettre aux équipes de répondre à toute dégradation de l'expérience client, rapidement et automatiquement.

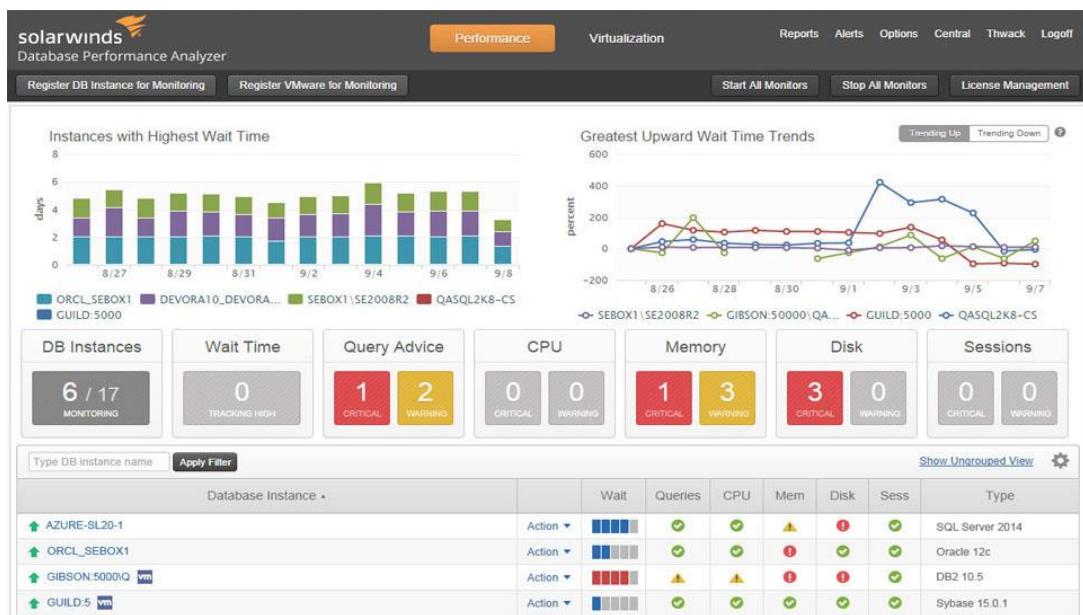


Figure 9: Tableau de bord d'un outil de monitoring, Solarwinds

L'enjeu pour les entreprises qui appliquent l'ensemble de ces concepts DevOps au quotidien est très grand. En effet, l'approche DevOps garantit un processus répétable et fiable qui limite les interventions manuelles. Ainsi, cette démarche accroît la productivité en permettant des déploiements automatisés plutôt qu'un ensemble de tâches manuelles.

Par conséquent, elle améliore la traçabilité et permet d'expliquer, de comprendre, de faciliter les diagnostics en cas de problème.

De plus, elle permet également des « rollbacks » (retour en arrière) grâce à la possibilité de remonter rapidement une infrastructure et d'offrir plus de « self-service » aux équipes de développement et décloisonner ainsi des actions jusqu'à présents réservées à la production.

[8]

On constate que ces dernières années, le DevOps séduit de plus en plus les entreprises et les industries. Selon une étude réalisée par Capgemini mi-2015 auprès de plus de 1500 professionnels dans 32 pays, plus de 50% des entreprises intègrent désormais les principes du DevOps à au moins la moitié de leurs projets de développement logiciel.

Cependant, la méthodologie DevOps ne se résume pas à de l'automatisation et nécessite une bonne intégration avec la culture de la communication et la collaboration entre les équipes pour qu'elle soit la plus efficace possible. En effet, l'approche DevOps doit également englober les parties prenantes d'un projet tels que le marketing, la direction et même les clients.

8.3. Avantages / Inconvénients / Limites

Le mouvement DevOps rassemble, et continue d'intégrer, de nombreux principes et bonnes pratiques pouvant être adoptés par des organisations IT de toutes tailles. Toutes ces expériences ont créé une approche qui vise à améliorer la façon dont l'IT apporte de la valeur ajoutée à ses clients. [10] Nous allons désormais nous intéresser aux avantages, inconvénients et limites que présentent cette méthodologie de travail.

Avantages de la méthode DevOps

- **Accélération du délai de mise sur le marché de nouveau produit**

L'un des avantages inhérents du DevOps est qu'il accélère la fréquence et la vitesse à laquelle les entreprises peuvent introduire des nouveaux produits sur le marché afin de maintenir un avantage concurrentiel. Plus une entreprise publie rapidement de nouveaux produits et ses mises à jour logiciels, plus vite elle pourra profiter de la valeur commerciale des fonctionnalités du produit.

Ainsi, l'adoption d'une démarche DevOps accélère le TTM (Time-to-market) grâce notamment aux tests continus et à l'automatisation. Il permet aux équipes de garder un œil sur le produit tout au long de son cycle de vie pour toute mise à jour logicielle. Cela réduit le temps de surveillance, de localisation et correction des bogues, ce qui a pour effet de réduire le TTM. [11]

- **Automatisation**

L'automatisation des tâches est également l'un des avantages clés de la méthode DevOps. En effet, cela évite les erreurs dues aux saisies manuelles ou manipulations et oblige à réfléchir sur les processus et les rôles de chacun. Lors d'un déploiement, l'utilisation d'un outil de gestion de configuration automatisé garantit l'uniformité entre les différents environnements et gère automatiquement les configurations propres à un environnement (configurations différentes entre la production et la non-production).

L'automatisation permet d'éliminer les erreurs humaines qui peuvent survenir dans les opérations manuelles et réduit le temps consacré aux tâches routinières. [12]

- **Application de concepts de qualité**

D'autre part, avec l'application des concepts DevOps notamment la surveillance continue, les équipes Ops sont capables de résoudre les problèmes plus rapidement. En effet, les outils de monitoring surveillent un ensemble de métriques définies préalablement pouvant être par exemple l'usage de la mémoire d'un serveur ou encore le nombre de connexions en simultanées sur une application.

Grâce à l'analyse de ces métriques sur ces outils, on peut rapidement être alerté si un seuil est dépassé et éviter des dysfonctionnements en anticipant un grand nombre de problèmes. D'autres concepts, comme les tests continues, garantissent une certaine stabilité du code.

En effet, lorsqu'une release est déployée sur un environnement d'exploitation, elle a forcément validé les phases de tests. Si les tests ont été bien définis et traitent un très grand nombre de cas pouvant assurer la qualité de ce qui est déployé, on peut garantir que les environnements d'exploitation gérés par des outils DevOps assurent une certaine stabilité.

D'après une enquête réalisée par Puppet Labs, les organisations performantes perdent 22% de temps en moins sur le travail et les changements imprévus. Cela leur permet de consacrer 29% de temps en plus aux tâches nouvelles, comme le développement de nouvelles fonctionnalités ou de nouveau code. Voilà pourquoi on en vient à en tirer un autre avantage d'adopter le DevOps : l'innovation.

- **Innovation**

Comme évoqué précédemment, le DevOps permet de livrer des produits logiciels rapidement. Cette vélocité libère une partie du temps des équipes, afin qu'ils puissent le consacrer pour expérimenter des fonctionnalités supplémentaires, améliorer l'efficacité des fonctionnalités et des infrastructures existantes en validant leurs faisabilités à l'aide de POC avec une perturbation minimale du projet en cours. Ainsi, le DevOps nourrit l'innovation en permettant aux équipes d'en savoir plus et de mieux comprendre les attentes des clients. [\[11\]](#)

Inconvénients / Limites de la méthode DevOps

Le DevOps encourage la collaboration immédiate entre les équipes informatiques, ce qui est tout à fait logique en théorie mais beaucoup plus difficile à mettre en œuvre, en particulier pour les entreprises établies depuis longtemps. Il s'agit généralement d'un mélange de problèmes culturels, liés aux tâches et technologiques.

- **Tâches**

Dans les entreprises qui existent depuis un très grand nombre d'années, les directeurs du développement sont confrontés à la réalité suivante : la grande majorité des activités sont axées sur la maintenance régulière ou de routine ou sur des améliorations modestes. La part du budget informatique consacrée aux activités de maintenance est généralement estimée à entre 70 et 80%.

Par conséquent, la majorité des activités de développement ne partent pas de zéro mais nécessitent une analyse détaillée du code actuel, un débogage et des tests unitaires approfondis et un investissement considérable en termes d'intégration et de test suite aux modifications.

La plupart de ces tâches se basent sur des workflows fixes et séquentiels qui peuvent difficilement être accélérées, en particulier étant donné les restrictions en matière d'utilisation des technologies. [\[13\]](#)

- **Technologies**

Les investissements en technologies et en architecture habituellement consentis dans les plus grandes entreprises ont souvent du mal à suivre le rythme des niveaux d'investissement dans d'autres domaines. C'est pourquoi des outils des années 1970 et 1980 sont toujours utilisés en tant que plateforme pour le développement de systèmes. Ces outils étaient certainement parfaitement adaptés à l'époque, mais ils sont bien loin de ce qui est possible en termes d'efficacité du développement moderne et de collaboration entre groupes. [\[13\]](#)

Le DevOps est une approche très prometteur en tant que modèle conceptuel, mais exige beaucoup de structures et fondements actuels, sans quoi sa mise en œuvre est compromise.

- **Sécurité**

Jusqu'à maintenant on a vu que l'approche DevOps permettait d'avoir des gains en termes de délai de développement et de qualité de code, mais qu'en est-il de la sécurité ?

Une des limites que présente cette approche est que par défaut elle ne prend pas en compte la composante sécurité au sein de la boucle DevOps. Auparavant, les processus liés à la sécurité étaient isolés dans une équipe spécifique lors de l'étape finale du développement. Cela ne posait pas de problème à une époque où les cycles de développement duraient des mois, voire des années. Mais cette époque est révolue.

Si une approche DevOps efficace garantit des cycles de développement rapides et fréquents (parfois quelques semaines ou jours), des pratiques de sécurité dépassées peuvent réduire à néant les bénéfices des projets DevOps les plus efficaces. [\[14\]](#)

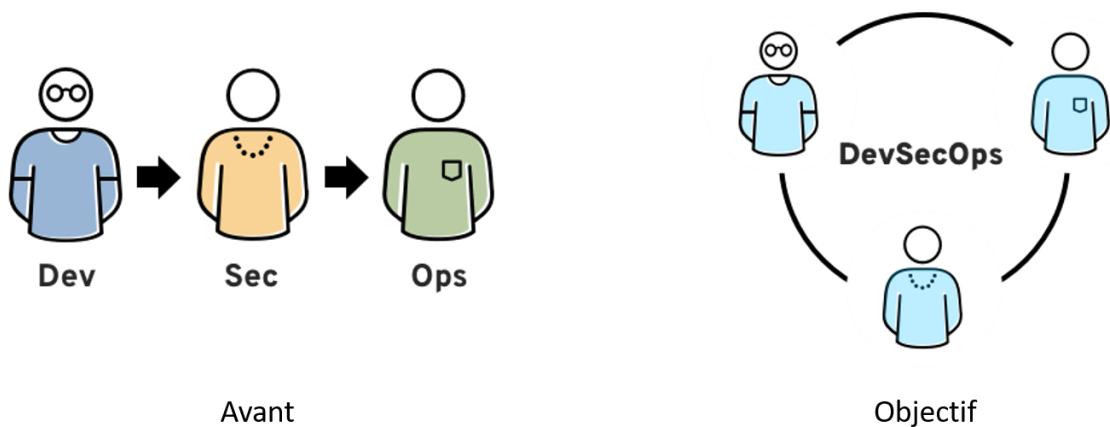


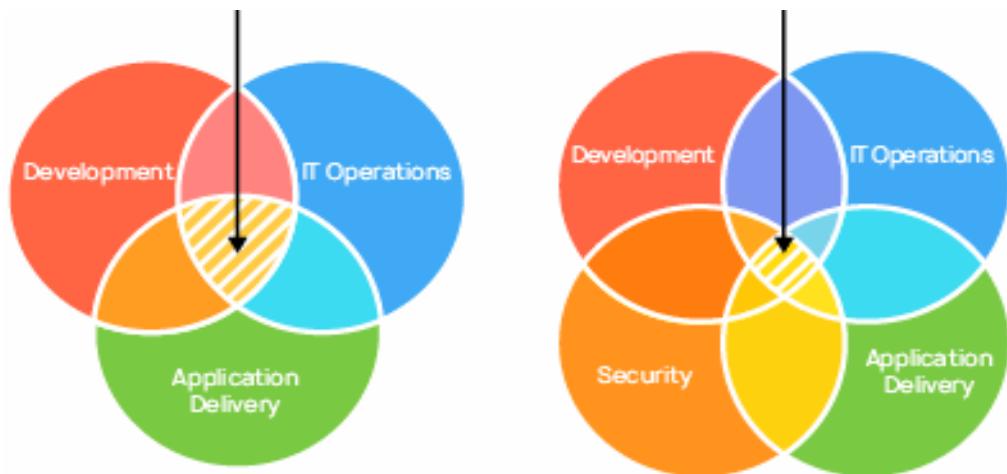
Figure 10: Evolution de la sécurité au sein de DevOps

Afin de conserver l'ensemble des gains issues des pratiques DevOps, il faut considérer la sécurité comme une responsabilité partagée, intégrée du début à la fin, ce qui a donné naissance à la notion de DevSecOps.

9. DevOps + Sécurité = DevSecOps

9.1. Qu'est-ce que le DevSecOps ?

Le DevSecOps (Development – Security – Operations) est une approche qui permet d'intégrer la sécurité des données dès le début d'un projet (*secured by design*). La sécurité de celles-ci est considérée comme une condition préalable avant de commencer. L'objectif est de pouvoir l'intégrer dans l'ensemble du cycle de vie du projet, du développement à la mise en œuvre, en utilisant des méthodes flexibles et l'approche DevOps. [15]



Avec l'émergence des cyberattaques, il est aujourd'hui nécessaire de se prémunir contre le vol de données ou encore les intrusions. L'objectif d'une telle démarche est de pouvoir mettre en place des processus de tests automatisés, et ce à chaque étape du projet, mais également au niveau de l'infrastructure. Avec une approche de type DevSecOps, la sécurité devient une responsabilité partagée et intégrée du début jusqu'à la fin d'un projet DevOps.

A travers la mise en place d'une approche DevSecOps, on cherche à [14] :

- **Conserver** des cycles de développements courts et fréquents ;
- **Intégrer** des mesures de sécurité aux processus d'exploitation avec un minimum d'interruptions ;
- Rester en phase avec les **technologies novatrices** telles que les conteneurs et les microservices ;
- **Encourager une collaboration** plus étroite entre les équipes habituellement isolées.

L'application de l'ensemble de ces initiatives permettrait à une entreprise à continuer à développer et livrer de nouvelles fonctionnalités de manière fréquentes tout en garantissant la sécurité de son application.

9.2. L'importance d'intégrer la sécurité au cœur du DevOps

Ces dernières années, on a constaté que la cybercriminalité évoluait à un rythme effréné, et que de nouvelles tendances ne cessaient d'apparaître. En effet, les cybermalfaiteurs acquièrent une plus grande aisance, adoptent les nouvelles technologies à la vitesse de l'éclair, adaptent leurs attaques en recourant à de nouvelles méthodes et coopèrent entre eux par des moyens inédits de notre point de vue.

Des réseaux criminels complexes agissent à travers le monde, en coordonnant en quelques minutes des attaques très élaborées. [\[Interpol\]](#) De plus, avec l'apparition de la pandémie du coronavirus, on remarque dans la figure ci-dessous que les entreprises sont encore plus en danger.

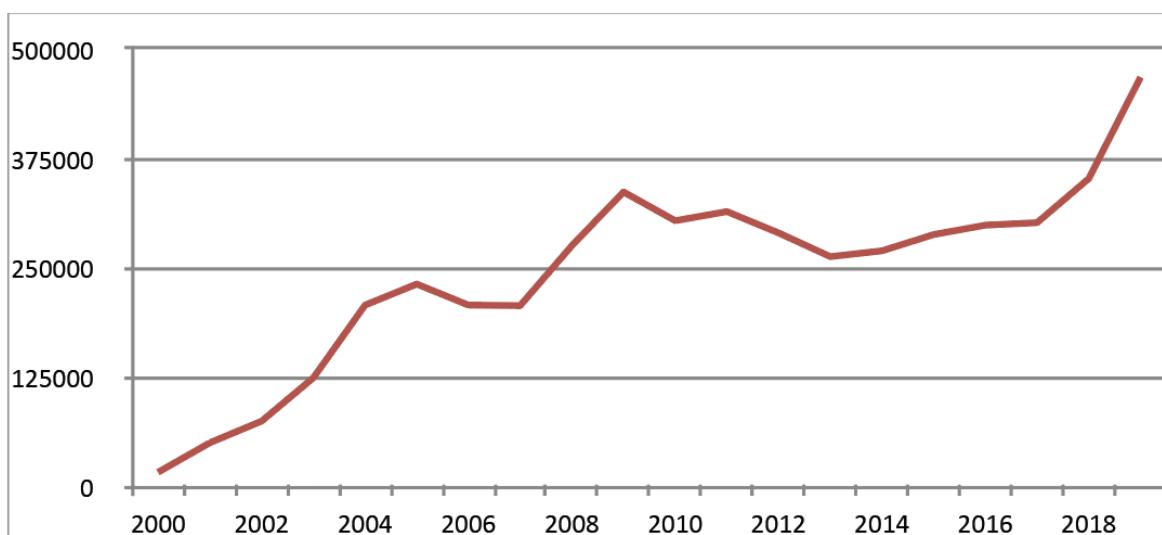


Figure 12: Evolution de la cybercriminalité dans le monde

Face aux attaques de sécurité, rançongiciels et fuites de données massives, préjudiciables à l'images des entreprises, les DSI doivent désormais élaborer une stratégie de sécurité informatique proactive basée sur le « Security by Design » et la gestion des risques.

Des recherches ont montré que 58% des entreprises avaient subis une violation des données, et 41% d'entre elles provenaient de vulnérabilités logicielles. Les erreurs de sécurité peuvent causer des dommages considérables et coûter des millions aux organisations. [\[16\]](#)

De plus, l'impératif de confidentialité et de sécurité dès la conception est devenu urgent à la suite de l'introduction du RGPD en 2018, qui a apporté des mesures de protection des données beaucoup plus strictes et un plus grand accent sur la responsabilité et la transparence.

Grâce au DevSecOps, avec qui la sécurité qui devient une composante naturelle du processus de développement, ces exigences sont prises en compte naturellement. Il est également plus facile et moins coûteux d'intégrer des mesures de sécurité dans le logiciel dès le début et, en prévenant les violations sur toute ligne, pour obtenir à la fois une sécurité améliorée et une satisfaction des clients garantie. [\[17\]](#)

Par ailleurs, l'intégration de la sécurité au cœur du DevOps permettrait également d'améliorer la manière de gérer le cycle de vie d'un projet notamment en appliquant les pratiques suivantes [17] :

- **L'analyse de code** : produire le code par petits bout pour faciliter la détection des vulnérabilités plus rapidement et le plus tôt possible ;
- **La gestion du changement** : augmenter la vitesse et l'efficacité en permettant aux membres de l'équipe de soumettre des changements ou des évolutions, puis déterminer si le changement peut être intégré ;
- **La surveillance de la conformité** : se mettre en tête lors du développement que le code sera audité dans la foulée et rester conforme aux frameworks et aux règles de développements établies ;
- **La quête des menaces et des vulnérabilités** : identifier les menaces potentielles en développement dans chaque mise à jour ou grâce à l'analyse du code et régit très rapidement pour les corriger. Des outils de sécurité adaptés aux équipes DevOps existent et sont capables d'identifier le code vulnérable au fur et à mesure de son écriture, car cela facilite une action immédiate pour les corriger ;
- **La formation à la sécurité** : investir dans la formation des développeurs sur la sécurité des applications entraînerait des avantages à long terme dans la mise en place d'une culture DevOps sécurisée.

C'est pourquoi il apparaît comme indispensable de gérer la composante sécurité au sein du cycle de vie de ces applications. Cependant, malgré les nombreux avantages que peuvent présenter cette approche, elle n'est pas forcément simple à mettre en œuvre.

9.3. Avantages / Inconvénients

Avantages du DevSecOps

- **Automatisation et contrôle continu**

Pour éviter tout ralentissement des flux DevOps et puisque l'exécution de contrôles de sécurité manuels peut être laborieuse et coûteuse en temps, l'automatisation des tâches répétitives est un élément clé de l'approche DevSecOps.

L'automatisation s'applique notamment au contrôle des développements : les développeurs peuvent tester en continu leur code pour identifier au plus tôt les éventuelles vulnérabilités et réduire le nombre de correctifs post-déploiement.

Elle touche également le contrôle des systèmes à travers la conteneurisation des solutions qui permet d'isoler les différentes fonctions d'un système, d'automatiser les opérations d'audit de sécurité et de vérifier à chaque instant que les politiques de cybersécurité sont bien mises en application.

L'utilisation d'environnements conteneurisés permet en outre de sécuriser l'infrastructure en automatisant la détection des incidents. Ainsi, lorsqu'une tentative d'intrusion ou un flux anormal est détecté, il est possible de désactiver et d'isoler les instances corrompues et de rediriger instantanément le trafic. [\[18\]](#)

L'automatisation permet également aux équipes de développeurs de pouvoir revenir à des versions antérieures rapidement en cas de failles de sécurité liées au code.



Figure 13: DevOps + Automation = Security

Enfin, plus un bug est détecté tardivement dans le cycle de vie, plus son coût de résolution est élevé. En effet, un bug de sécurité repéré en production coûte 100 fois plus cher à résoudre que lorsqu'il a été détecté dans la phase de spécification.

En automatisant les tests de sécurité sur l'ensemble du cycle de vie applicatif, le coût potentiel de non qualité pour l'entreprise sera infiniment moindre. De plus, le traitement des vulnérabilités de façon intégrée permet de ne pas ralentir le rythme de livraison. (Annexe 2)

- **Priorisation de la sécurité**

L'approche DevSecOps repose en effet sur une sécurité intégrée, et non sur un périmètre de sécurité protégeant les applications et les données. L'inconvénient de traiter la sécurité hors du cycle de développement DevOps est qu'il entraîne les entreprises à être confrontées à de longs cycles de développement, ce qu'elles essaient précisément d'éviter.

Pour remédier à cela, il faut réfléchir à la sécurité de l'application dès les phases de conception, ce qui signifie d'avoir une collaboration entre les équipes de développement et celles dédiées à la cybersécurité. On constate que cette approche met en avant la composante sécurité en établissant un cadre de référence pour l'ensemble des activités de développement.

Les tests de sécurité doivent être effectuées à tous les niveaux, c'est-à-dire les environnements de développement, de qualification, de test et de préproduction. La surveillance automatisée du résultat de ces tests alertera en temps réel les équipes concernées sur les vulnérabilités bien avant que le produit final ne soit livré en production.

[\[15\]](#)

- **Respect des exigences de sécurité / conformité**

D'après l'étude réalisée par Micro Focus (Annexe 2), 82% des répondants jugent que l'un des avantages d'une démarche 100% DevSecOps est de pouvoir respecter les exigences de sécurité et de conformité.

En effet, les entreprises qui ne seraient pas conformes aux nouvelles réglementations comme la RGPD se verraient infliger de lourdes sanctions, comme on le voit sur la figure ci-dessous, pouvant aller jusqu'à 4% de leur chiffre d'affaires annuel.

Les sanctions du RGPD



Figure 14: Sanctions du RGPD

Par exemple, la CNIL a infligé une sanction de 50 M€ à Google pour manquement à ses obligations au règlement général européen de protection des données. C'est donc dans le but d'éviter de lourdes sanctions mais également de se protéger d'attaques comme des fuites de données que les entreprises songent à mettre en place les pratiques DevSecOps.

- **Développement d'une culture de sécurité**

Pour que l'approche DevSecOps soit efficace, une coopération étroite est nécessaire entre les développeurs, les équipes opérationnelles, le service informatique et les parties prenantes du projet. Par conséquent, afin de développer une culture de la sécurité, il est nécessaire de créer un climat de confiance entre les groupes.

Afin que chacun se sente impliqué dans la coopération, des formations et des sessions de sensibilisation à la sécurité peuvent être planifiées. Le DSI joue un rôle important car il a une vue d'ensemble du projet et des équipes et permet de jouer un rôle transverse, nécessaire au bon fonctionnement du projet. [15]

Au-delà des outils de protection et des mails bienveillants de bonnes pratiques CNIL ou ANSSI, la clé de la réussite en matière de cybersécurité se résume en un mot : l'humain. Sensibiliser et former ses équipes aux risques informatiques est une clé dans l'optique d'avoir une cybersécurité performante. Comme on le voit dans la figure ci-dessous, l'efficience d'une entreprise au niveau de la sécurité repose sur 3 piliers : la sécurité technique, les systèmes de management ainsi que les facteurs organisationnels et humains.

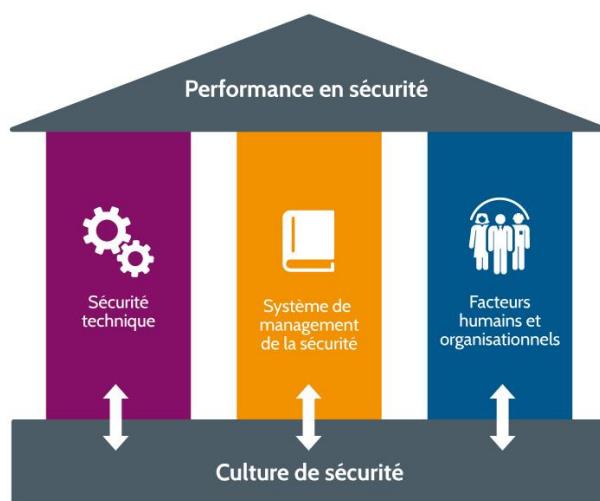


Figure 15: Les 3 piliers de la sécurité

En outre, lorsque les différentes questions de sécurité sont bien intégrées dans le processus de développement, les équipes sont capables de produire des versions de logiciel plus sûres et stables et ainsi permettre à l'entreprise de livrer de nouvelles fonctionnalités aux clients très fréquemment.

Inconvénients / Limites du DevSecOps

Comme pour l'approche DevOps, la réussite du système et son efficacité dans le cadre d'une approche DevSecOps dépendent de la capacité des collaborateurs et des équipes à gérer ce nouveau style de développement. [19]

C'est pourquoi, il est nécessaire de bien communiquer sur les avantages du nouveau système, mais aussi de bien coordonner les changements avec les services et les collaborateurs, afin d'éviter l'émergence de complications.

D'après l'enquête réalisée par Micro Focus (Annexe 2), on constate que plusieurs points se présentent comme des freins pour les entreprises dans leur mise en place d'une démarche DevSecOps.

QUELS SONT LES FREINS À UNE DÉMARCHE 100% DEVSECOPS ?

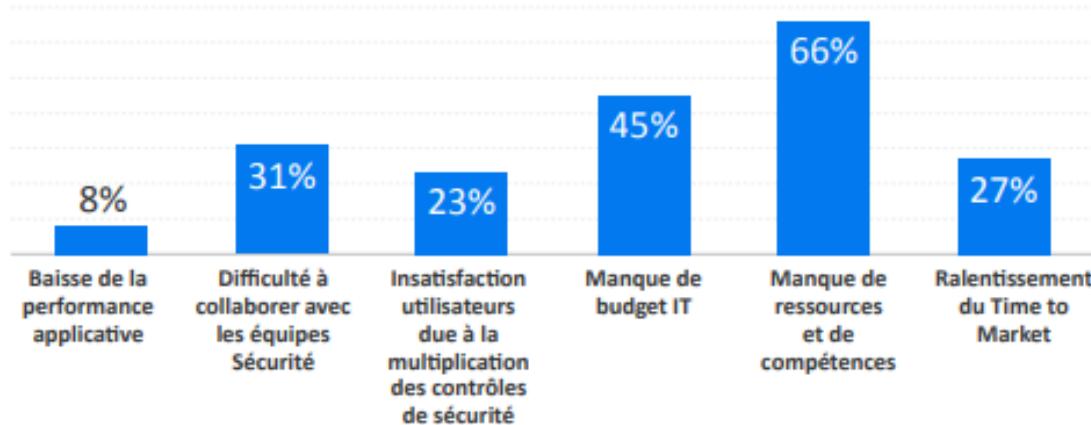


Figure 16: Freins à la démarche DevSecOps

Parmi ces points, le manque de ressources et de compétences représente 66% des réponses et le manque de budget 45%. On en conclut d'après cette enquête que les équipes agiles ne sont pas assez sensibilisées aux bonnes pratiques de sécurité informatique et qu'il faudrait assurer des formations afin d'avoir des équipes qui ont les compétences pour assurer une efficacité maximale dans cette démarche.

De plus, le temps consacré dans la définition des exigences de sécurité reste beaucoup trop minime pour qu'elles permettent d'assurer la sécurité des applications développées.

Malgré ces inconvénients, l'approche DevSecOps présente un très grand nombre d'avantages qui permettrait à une entreprise de conserver les bénéfices d'un développement DevOps tout en assurant la sécurité de ce qu'elle fournit.

On remarque que les initiatives DevSecOps semblent s'installer progressivement dans les entreprises. En effet, près de la moitié des répondants (41%) ont déclaré qu'ils allaient investir dans cette voie à court et moyen terme, ce qui atteste d'une nouvelle prise en considération et de sa légitimité pour la mise en conformité.

AVEZ-VOUS UN PROJET D'INVESTISSEMENT DEVSECOPS ?

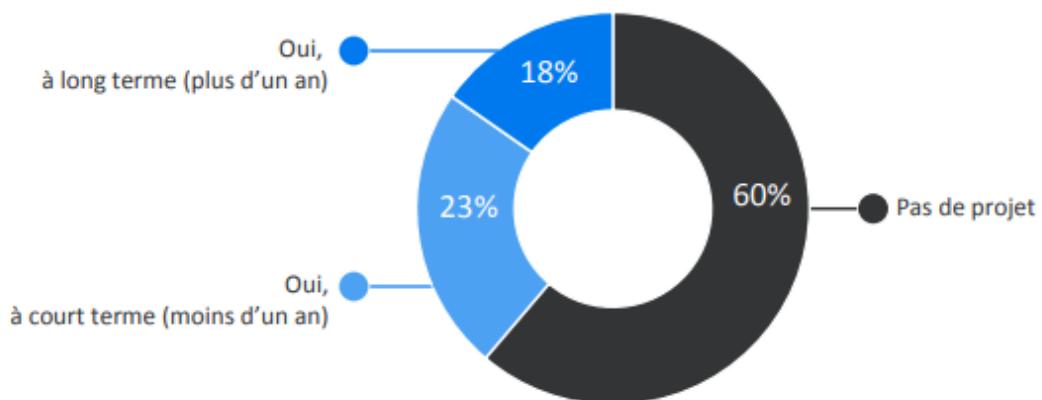


Figure 17: Projet d'investissement DevSecOps

Dans la partie suivante, nous allons nous intéresser à savoir comment mettre en place l'approche DevSecOps et quels sont les bonnes pratiques à suivre afin de tirer au maximum l'ensemble des bénéfices de cette philosophie au service de son entreprise.

10. Mise en place d'un modèle de développement DevSecOps

10.1. Plan d'action de mise en place du DevSecOps

10.1.1. Intégration de la sécurité dans la boucle DevOps

Comme on l'a vu précédemment, dans l'approche DevOps on travaillait en parallèle avec une équipe sécurité afin d'appliquer et de garantir la sécurité des applications. Mais de ce fait, le cycle de vie de l'application perdait les avantages procurés par le DevOps de pouvoir déployé et livré du code plus fréquemment avec des cycles itératifs plus fréquents.

Pour remédier à cette pratique, il faut intégrer la sécurité au sein de la boucle DevOps et rendre l'application « **Secured By Design** ». Pour cela, il y a des contrôles de sécurité spécifiques qui peuvent être appliqués à chaque phase :

- **Plan**

Il est important de s'interroger sur la manière dont les tests de sécurité vont être intégrés au sein du cycle de vie de l'application. En effet, dans l'étape de planification, il faut créer des plans de tests afin de déterminer l'ensemble des scénarios possibles, la fréquence de lancement des tests et l'outillage qui sera utilisé pour répondre aux différents enjeux de sécurité auquel l'entreprise doit faire face.

- **Code**

Dans la phase de développement, il faut penser à intégrer des outils de sécurité et des plugins directement dans l'IDE, ce qui permettra d'identifier toute vulnérabilité du code source. De plus, des outils de *linting*, c'est-à-dire des outils permettant l'analyse statique du code source et étant capable de détecter des potentiels problèmes de syntaxes dans le code et proposer des solutions, peuvent être ajoutés.

The screenshot shows a code editor with the following JS code:

```
1 'use strict';
2 ● var foo = "bar";
3
4 ● fn(function (err) {});
```

Annotations from a static analysis tool:

- Line 2, Col 5: Error: foo is defined but never used (no-unused-vars)
- Line 2, Col 11: Error: Strings must use singlequote. (quotes)
- Line 4, Col 1: Error: "fn" is not defined. (no-undef)
- Line 4, Col 4: Warning: Expected error to be handled. (handle-callback-err)
- Line 4, Col 14: Error: err is defined but never used (no-unused-vars)

Figure 18: Exemple de *linting* de code JS

L'ensemble de ces outils font parti d'une méthode de test de sécurité particulière : la méthode **SAST** (Static Application Security Testing). Cette dernière a pour objectif d'identifier les faiblesses présentes dans le code source en lançant des tests qui vont scanner l'intégralité du code. D'autres outils SAST existent dont la spécialité est de rechercher automatiquement des secrets qui sont stockées dans le code (clés privés, mots de passe, clés de session, etc).

- **Build**

Durant la phase de Build, une autre méthode de test de sécurité peut être mise en place : la méthode **SCA** (Software Composition Analysis). Les outils de cette méthode ont pour objectif de vérifier la composition d'une application en termes de dépendances tierces et de licences. En effet, la plupart des applications embarquent des frameworks ou encore des bibliothèques Open Source. Il est donc indispensable de vérifier que l'artefact qui résulte du build du code source n'embarque pas des composants connus comme vulnérables ou obsolètes mais également qu'il n'y ait pas de défaut de compatibilité de licences. [\[20\]](#)

- **Test**

Dans la pratique DevSecOps, les contrôles de sécurité les plus importants sont appliqués dans la phase de Test. En effet, à ce stade du pipeline, on peut utiliser des outils de type **DAST** (Dynamic Application Security Testing) pour assurer la sécurité de son application. En tant qu'outils effectuant ce que l'on appelle les tests en boîte noire, les analyseurs dynamiques peuvent identifier les vulnérabilités des programmes telles que les injections SQL, les débordements de tampon, etc. Ces solutions de tests dynamiques de sécurité des applications ont donc pour objectif de détecter des vulnérabilités et des faiblesses d'une application au cours de son exécution, notamment les applications Web.

La méthode DAST peut également mettre en lumière les problèmes d'exécution qui ne peuvent pas être identifiés par une analyse statique, par exemple les problèmes d'authentification et de configuration du serveur, ainsi que les failles visibles uniquement lorsqu'un utilisateur connu se connecte.

- **Release**

Lors de la phase de Release, il est recommandé d'appliquer des outils de sécurité qui vont effectuer des tests de pénétration approfondis dans l'application et également réaliser une analyse des vulnérabilités présentes dans le code. Pour cela, on peut utiliser les outils de type **RASP** (Run-time Application Security Protection) qui vont se brancher à l'application ou à un environnement d'exécution particulier et contrôler l'exécution. Les outils RASP protègent l'application même si les défenses du périmètre réseau sont violées et que les applications contiennent des failles de sécurité qui n'ont pas été détectées par l'équipe de développement.

RASP permet à une application d'effectuer des contrôles de sécurité continus sur elle-même et de répondre aux attaques en direct en mettant fin à la session de l'attaquant et en alertant les défenseurs de l'attaque.

- **Deploy**

Enfin, afin d'assurer la sécurité de l'application en production, on peut intégrer des outils tels qu'un **WAF** (Web Application Firewall) qui va protéger le serveur d'applications Web contre de multiples attaques (phishing, ransomware, attaque DDOS, malware). La fonction du WAF est de garantir la sécurité du serveur Web en analysant les paquets de requête HTTP / HTTPS et les modèles de trafic. (Figure 19)

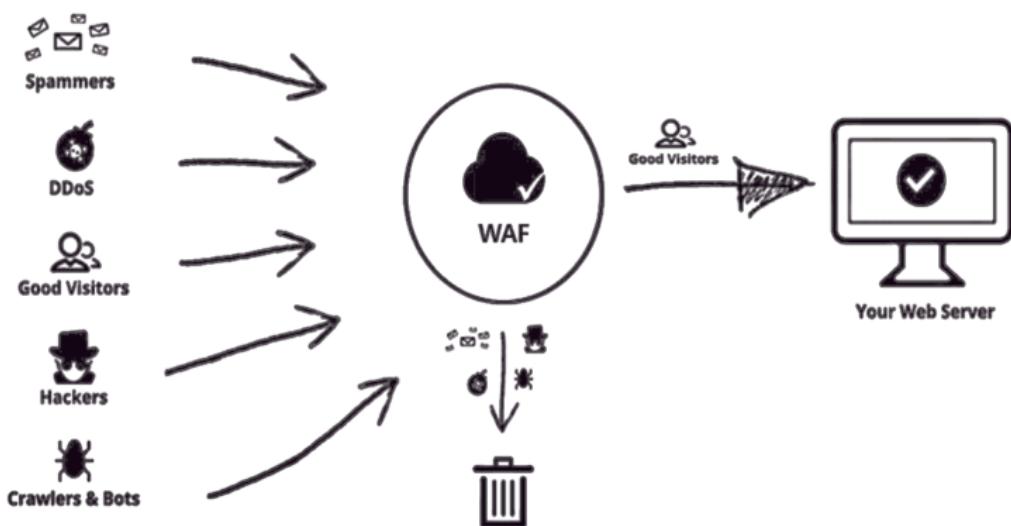


Figure 19: Fonctionnement d'un WAF

Le WAF examine chaque demande envoyée au serveur, avant qu'elle n'atteigne l'application, de manière à vérifier que cette demande soit en conformité avec les règles du pare-feu. Les fonctionnalités du WAF peuvent être implémentées :

- En software : une application est installée sur le système d'exploitation
- En hardware : les fonctionnalités sont intégrées dans une solution d'appliance.

Le WAF permet également de répondre aux exigences de conformité, éliminer la complexité de la gestion et réduire les coûts d'exploitation pour toutes les entreprises ayant des applications et des services Web orientés Internet. [\[21\]](#)

Dans la figure suivante, on voit un schéma synthétique décrivant les pratiques de sécurité au sein de la boucle DevOps.

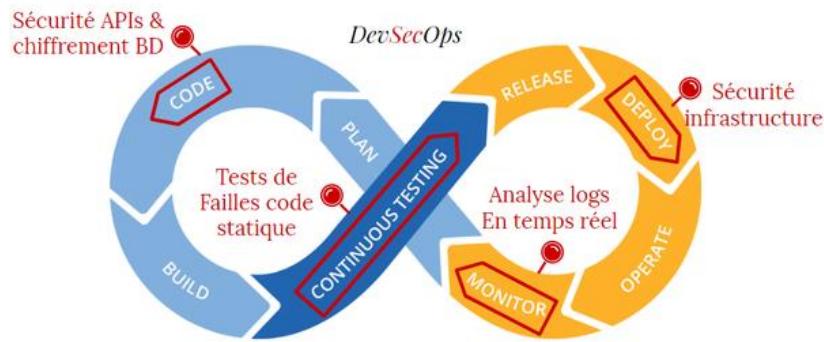


Figure 20: Pratiques de sécurité au sein de la boucle DevOps

10.1.2. Etapes à mettre en œuvre pour instaurer cette nouvelle méthodologie

Maintenant que nous connaissons les différentes méthodes de sécurité à ajouter à chaque étape de la pipeline DevOps ainsi que leurs avantages, nous allons voir en détail les étapes à suivre en tant qu'entreprise afin d'intégrer la sécurité du DevOps au sein de son organisation.

L'un des points clés du DevSecOps est la capacité à pouvoir **établir des rôles avec des responsabilités bien distinctes**. En effet, on doit être capable de reconnaître clairement :

- Les développeurs, qui doivent se focaliser sur le développement d'applications et de logiciels ;
- Les équipes opérationnelles, qui ont pour objectif de gérer le développement d'une infrastructure stable, fiable et évolutive ;
- Les équipes de sécurité, qui ont pour rôle de gérer la protection des actifs et des données et la réduction des risques.

Les interactions entre chacun de ces groupes peuvent être codifiées dans une politique de sécurité préalablement définie. Par exemple, les développeurs peuvent mettre en place une politique de sécurité dans laquelle ils déclarent les priviléges requis par leur application ou leur service. L'équipes de sécurité se charge de passer en revue l'ensemble de ces points, puis les équipes Ops s'assurent que le déploiement de l'application se déroule comme prévu.

Il faut également penser à **intégrer la sécurité dans les pratiques CI/CD**. Dans le contexte DevOps, la sécurité est trop souvent considérée comme un aspect secondaire, dont on ne prend en compte généralement à la fin de l'ensemble des tâches définies. Les changements de dernière minute requis pour gérer les vulnérabilités repoussent la livraison des nouvelles versions de logiciels. Les entreprises innovantes utilisent aujourd'hui des outils de planification et de gestion du workflow comme Kanban pour modéliser les flux, accélérer le développement et éliminer les inefficacités.

En outre, les équipes de sécurité déconstruisent de plus en plus les applications en micro-services afin de simplifier les contrôles de sécurité et les changements à apporter.

Par ailleurs, l'une des étapes primordiales dans l'intégration de la sécurité dans DevOps est **l'adoption d'une approche proactive de la sécurité**. Des pratiques de sécurité robustes, tels que le contrôle d'accès aux outils DevOps ou encore le monitoring, doivent être mises en place tout au long du cycle de vie de l'application dans le but de réduire les vulnérabilités mais également d'améliorer la posture de sécurité et de réduire les risques. Les bonnes pratiques de sécurité de base pour le DevSecOps incluent les aspects suivants :

- Traiter les exigences de sécurité et les vulnérabilités potentielles de façon **holistique**, car il suffit parfois aux attaquants d'une seule vulnérabilité pour poursuivre leur mission ;
- **Réduire la concentration des privilèges** dans les outils d'automatisation des versions et s'assurer que les référentiels de code n'exposent aucun secret (clé privée, mot de passe) ;
- Conserver les secrets utilisés par les machines et les personnes (mots de passe, certificats, clés API, tokens et clés SSH) dans un **coffre-fort sécurisé et hautement disponible**, en dehors du code source et en dehors des ordinateurs des développeurs et des systèmes de stockage accessibles par les utilisateurs. Renouveler régulièrement les secrets pour réduire l'exposition ;
- Appliquer le principe du **moindre privilège** pour garantir que les machines et les personnels peuvent uniquement accéder aux ressources dont elles ont besoin ;
- **Etablir une base de référence** des modèles d'utilisation normaux afin de détecter les anomalies de telle sorte que les utilisateurs mal intentionnés puissent être retrouvés sans qu'ils ne volent d'identifiants ;
- **Attribuer à chaque machine sa propre identité** unique afin d'auditer et de superviser la façon dont elle accède aux différents secrets ;
- Exécuter des **analyses de vulnérabilité** et des **tests de pénétration** pour détecter les éventuels failles et les corriger.

Enfin, afin de conserver les bénéfices de l'approche DevOps, il est primordial d'**automatiser les processus de sécurité**. En effet, la méthodologie DevOps utilise l'automatisation pour accélérer la gestion du cycle de vie des applications et éliminer la latence des opérations humaines.

De même, la sécurité du DevOps doit s'appuyer sur **l'automatisation** pour réduire au maximum les interactions humaines et manuelles. Par exemple, le renouvellement automatique des secrets (mots de passe, clé, certificats) permet aux entreprises d'éviter que les attaquants n'accèdent aux outils DevOps, aux clés d'accès ou encore aux systèmes pendant longtemps.

Il est également possible d'utiliser des procédures de sécurité automatisées en cas de détection d'une faille. Par exemple, les sessions à privilèges peuvent être automatiquement fermées et les identifiants automatiquement renouvelés dès lors qu'une faille de sécurité a été identifiée.

[22]

10.1.3. Autres points à prendre en considération

Le DevSecOps est plus qu'une méthodologie, c'est une **culture** à adopter pour les entreprises, qui peut se résumer en trois mots : **Confiance, Responsabilité et Coopération**.

La **confiance** est primordiale pour combler le fossé qui s'est formé entre le DevOps et la sécurité. Les entreprises doivent mettre l'accent sur la dynamique entre développeurs et administrateurs et promouvoir une culture qui respecte les équipes de développement, de production et contribuant à la confiance.

La **responsabilité** des développeurs dans le domaine de la sécurité est également un élément clé du DevSecOps. Avec la sensibilisation aux failles et aux menaces qui pourraient potentiellement nuire aux applications, et le dialogue avec l'équipe sécurité, cette responsabilité ne peut que se renforcer.

Enfin, la **coopération** entre les 3 équipes est la dernière dimension de la culture DevSecOps. Pour créer une équipe unifiée axée sur DevSecOps, l'équipe de sécurité doit travailler et collaborer avec les développeurs et les administrateurs.

L'équipe sécurité n'est pas dimensionnée pour accompagner chaque équipe de développement et de production sur les différentes problématiques de sécurité. Il est donc important d'avoir un correspondant de l'équipe de sécurité dans chacune de ces équipes afin d'assurer l'adhésion des développeurs et des administrateurs à la démarche DevSecOps : c'est ce qu'on appelle le **Security Champion**.

Le rôle de ce dernier est d'évangéliser la démarche DevSecOps et diffuser la culture sécurité auprès de son équipe. Il est responsable de :

- La contribution à la **conception sécurisée** de l'architecture applicative ;
- La veille au **respect des bonnes pratiques** de sécurité des développements ;
- La réalisation ou la vérification des **revues de code** ;
- La **rédaction d'user stories** orientées sécurité.

Pour aider les Security Champions à monter en compétence, l'entreprise ne doit pas hésiter à mettre en place :

- **Programme de formation** sur les sujets techniques et fonctionnels de la sécurité applicative ;
- **Coaching par un expert en sécurité informatique** présent au niveau de l'entreprise ;
- **Communauté** regroupant les Security Champions permettant l'entraide et l'autoformation.

[23]

10.2. Outils recommandés et nécessaires

La mise en place de cette nouvelle méthodologie nécessite de nombreuses piles technologiques avec plusieurs solutions qui doivent être soigneusement intégrées. Voici quelques outils DevSecOps importants et recommandés qui peuvent être mis en place par une entreprise :

SonarQube



Cet outil est utilisé pour l'inspection continue de la qualité du code. Il fournit une rétroaction continue sur la qualité des logiciels et supporte plus de vingt-cinq langages (Java, C, C++, Python, etc). Le scanner Sonar réalise une analyse sur l'ensemble de l'application et génère un reporting à la fin de l'analyse qui traite plusieurs points :

- Identification des duplications de code ;
- Mesure du niveau de documentation ;
- Respect des règles de programmation et des bonnes pratiques ;
- Détection des bugs potentiels ;
- Evaluation de la couverture de code par les tests unitaires ;
- [...]

ThreatModeler

Solution de modélisation automatisée des menaces qui évolue et sécurise le cycle de vie du développement logiciel de l'entreprise. Permet la prédiction, l'identification et la définition des menaces de sécurité permettant un gain de temps et d'argent.



Parmi les avantages de cet outil, on retrouve :

- Intégration transparente avec SSO ;
- Modèles d'architecture préconstruits pour démarrer rapidement ;
- Réutilisation du travail existant (importation de diagrammes existants) ;
- Bibliothèque de composants intégrée pour le Web, Cloud (AWS, Azure) ;
- Configuration et déploiement facile et rapide.

Aqua Security



Solution de prévention, détection et automatisation des réponses pour sécuriser la construction, sécuriser l'infrastructure cloud et sécuriser les charges de travail en cours d'exécution. En d'autres termes, Aqua Security sécurise l'ensemble du cycle de vie des applications.

L'outil Aqua Security analyse les artefacts à la recherche de vulnérabilités, de logiciels malveillants, de secrets et d'autres risques pendant le développement et la mise en place. Elle permet de définir des politiques flexibles et dynamiques pour contrôler le déploiement dans les environnements d'exécution.

Quant au niveau de l'infrastructure, Aqua est capable de vérifier les services Cloud, de scanner les modèles IaaS (Infrastructure As A Code) et la configuration Kubernetes par rapport aux bonnes pratiques et aux normes, afin de garantir que l'infrastructure sur laquelle est exécutée l'application est configurée de manière sécurisée et conforme.

CheckMarx

CheckMarx est un ensemble de solutions de sécurité logicielle qui fournit des tests de sécurité pour les applications statiques et dynamiques, des outils tels que l'analyse de composition logicielle et le code bashing pour promouvoir la culture de la sécurité logicielle parmi les développeurs.



Fortify



Ce dernier fournit la sécurité des applications en tant que service. En effet, il est principalement utilisé en entreprise pour le développement sécurisé, les tests de sécurité et la surveillance continue.

Fortify est un outil complet qui couvre les tests de types SAST, DAST, IAST et SCA. Il est capable de s'intégrer fortement au pipeline CI/CD notamment avec des fonctionnalités innovantes : audit assisté par le machine learning (Audit Assistant) et analyse en temps réel dans l'IDE grâce à Security Assistant. C'est également un outil flexible disponible on-premise ou en tant que service.

HashiCorp Vault

Ce dernier est un coffre-fort pour l'ensemble de vos données, c'est-à-dire qui va sécuriser l'accès à toutes données sensibles. Cet outil très complet permet également [\[24\]](#) :



- **Stockage secret sécurisé** : Vault crypte les secrets avant de les écrire dans le stockage persistant, donc accéder au stockage brut ne suffit pas pour accéder aux secrets ;
- **Secrets dynamiques** : l'outil peut générer des secrets à la demande pour certains systèmes, tels que les bases de données AWS ou SQL. Après avoir créé ces secrets dynamiques, Vault les révoquera également automatiquement une fois le travail terminé ;
- **Chiffrement des données** : Vault peut chiffrer et déchiffrer les données sans les stocker. Cela permet aux équipes de sécurité de définir des paramètres de chiffrement et aux développeurs de stocker des données chiffrées dans un emplacement tel qu'une base de données SQL sans avoir à concevoir leurs propres méthodes de chiffrement ;
- **Location et renouvellement** : l'ensemble des secrets sont associées à un bail. A la fin du bail, Vault révoquera automatiquement ce secret. Les clients peuvent renouveler les baux via des API de renouvellement intégrées ;

- **Révocation** : Vault prend en charge la révocation des secrets. L'outil peut révoquer non seulement des secrets uniques, mais aussi un arbre entier de secrets. Par exemple tous les secrets lus par un utilisateur spécifique, ou tous les secrets d'un type particulier.

GauntLT



GauntLT fournit une variété d'outils de sécurité et les met à la portée des équipes de sécurité, de développement et d'exploitation pour qu'elles collaborent à la création de logiciels robustes. Il est conçu pour faciliter les tests et la communication entre les groupes et créer des tests exploitables qui peuvent être intégrés à vos processus de déploiement et de test.

Les attaques de GauntLT sont écrites dans un langage facile à lire, ce qui peut faciliter le maintien de ce code dans le temps. De plus, ce dernier s'intègre facilement aux outils et processus de test de l'entreprise. Des adaptateurs d'outils de sécurité sont également fournis.

IriusRisk

Cet outil permet aux équipes de sécurité et de développement de collaborer, d'accélérer la mise sur le marché et de faire évoluer la sécurité plus facilement.



En utilisant IriusRisk au sein de votre entreprise, vous pouvez :

- Générer un modèle de menace en quelques minutes ;
- Identifier les menaces pour votre produit / logiciel et les contre-mesures ;
- Eviter les retards dans le déploiement et accélérer le temps de production ;
- Economiser du temps, des coûts et du travail de développement ;
- Crée une culture de collaboration entre les équipes de sécurité et de développement

[\[25\]](#)

La mise en place de l'ensemble de ces outils au sein de votre organisation permet de garantir la conservation des bénéfices provenant de l'approche DevOps tout en garantissant la sécurité des applications à chaque étape du pipeline. Ainsi, on assure la naissance d'une culture DevSecOps au sein de l'entreprise qui permettra une meilleure collaboration entre les équipes de développement, administration et de sécurité et qui permettra d'avoir des applications en production stables et sécurisées.

11. Application au sein de l'équipe DevOps de Dassault Systèmes

Dassault Systèmes a mis l'accent sur la sécurité avec l'augmentation du travail à distance dû à la montée de l'épidémie du coronavirus. De plus, avec l'augmentation du nombre d'attaques informatiques, il faut sécuriser au maximum l'ensemble des applications et logiciels DS et continuer à fournir la meilleure qualité de service pour les clients.

L'équipe DevOps est en charge de la livraison d'applications et de middlewares pour les autres équipes en interne. Parmi les améliorations de sécurité apportées, il y a eu la mise en place de communication chiffrée en HTTPS.

11.1. Chiffrement des communications SSL

En effet, de nombreuses communications entre des applications se faisant par des appels http, qui n'était pas sécurisé initialement et potentiellement vulnérable à des attaques de type man-in-the-middle. Par conséquent, les mots de passes qui étaient transmis dans ces requêtes circulaient en clair dans le réseau.

Pour remédier à cela, des outils tels que HA Proxy a été mis en place afin d'utiliser la fonctionnalité de *reverse-proxy*. Contrairement au serveur proxy qui permet à un utilisateur d'accéder au réseau Internet, le reverse proxy permet à un utilisateur d'accéder à des serveurs internes.

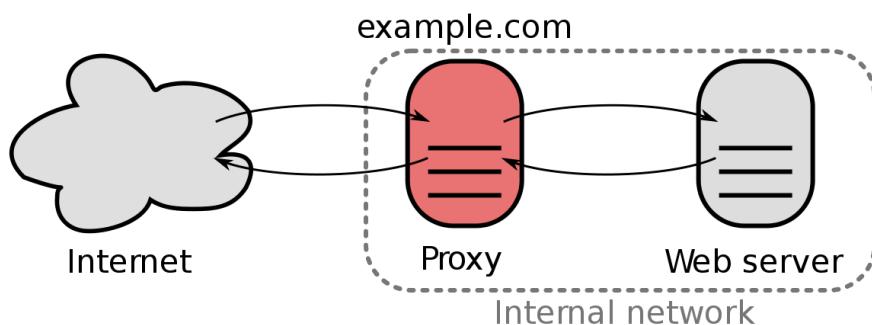


Figure 21: Principe de fonctionnement du reverse proxy

L'utilisation d'un reverse proxy possède plusieurs avantages comme la répartition de charge ou encore le chiffrement SSL. L'utilisation du chiffrement SSL permet de prendre en charge plusieurs principes de sécurités tels que la protection de la transmission de données, la garantie d'être connecté sur le bon serveur ou encore l'intégrité des données.

Aujourd'hui, avec la mise en place de cet outil, l'ensemble des communications sont chiffrées et complètement fiables entre les différentes applications gérées par l'équipe DevOps.

11.2. Mise en place de Remote Apps

Parmi les autres améliorations de sécurité réalisées dans l'équipe DevOps, il y a eu la mise en place des Remote Apps.

RemoteApp est une technologie délivrée par Windows qui permet de fournir des applications à distance. L'application ressemble à n'importe quelle autre application fonctionnant sur l'appareil de l'utilisateur final. Les serveurs RDS (Remote Desktop Services) permettent à des utilisateurs l'utilisation d'applications installées non pas sur leur machine mais sur leur serveur. Ainsi, il n'est pas nécessaire de déployer et de maintenir les applications sur tous les postes clients.

L'utilisation de cette technologie propose de nombreux avantages tels que :

- **Contrôle des accès aux applications**

Avec cette technologie, on peut facilement déclarer qui a accès ou non à un serveur d'applications et journaliser l'ensemble de l'activité d'un utilisateur sur une application. Pour ajouter un nouvel utilisateur, on a juste besoin de le déclarer dans une liste spécifique et l'accès lui est donné immédiatement.

- **Protection des données contre le vol informatique**

Etant donné que les données sont stockées sur un serveur distant et sécurisé dans un DataCenter, le vol d'un poste informatique d'un utilisateur n'impacte pas du tout la sécurité ou la production d'une entreprise car les données ne sont plus stockées en local.

- **Sauvegarde des données**

Les données des applications ou de l'utilisateur sont stockées au même endroit, ce qui rend aisée la sauvegarde ou la restauration de données depuis le serveur RDS. [\[26\]](#)

- **Utilisation du MFA (Multi-Factor Authentication) pour s'authentifier**

Pour intensifier le niveau de sécurité des serveurs RDS, l'authentification forte a été mise en place. En effet, pour pouvoir accéder à une application sur un serveur RDS, il faut faire partie d'un groupe qui a accès à cette application et ensuite être capable de fournir au minimum deux facteurs de vérification pour prouver son identité.

Cela peut être une combinaison entre :

- Connaissance de l'utilisateur comme un mot de passe ou encore un code PIN ;
- Possession de l'utilisateur qui peut-être son smartphone, ordinateur ou bien une carte à puce ;
- Inhérence de l'utilisateur, soit son empreinte digitale ou bien une reconnaissance vocale ou faciale

L'authentification multi-facteurs est ici principalement utilisée pour ajouter une défense supplémentaire et rendre plus difficile l'accès d'une personne non autorisée aux applications de DS.

12. Conclusion

Avec l'évolution constante de l'informatique, les entreprises se doivent d'être capable de mettre sur le marché de nouvelles fonctionnalités plus rapidement et de manière plus stable. En effet, les équipes doivent être capable d'ajouter de nouvelles fonctions sans aucune difficulté, de pouvoir corriger les bugs découverts et tout cela sans apporter de régression dans l'application en production. Et la solution pour répondre à ce besoin est l'approche DevOps.

Cette méthodologie permet une meilleure communication et collaboration entre les équipes de développement et d'exploitation et surtout un cycle de développement plus rapide et fiable et des déploiements plus efficaces qui permet à l'entreprise de réduire le TTM.

Cependant, avec l'augmentation du nombre de cyberattaques, notamment avec l'épidémie du Covid-19, les entreprises se doivent également de mettre l'accent sur la composante sécurité afin d'éviter toutes attaques informatiques et d'améliorer la sécurité de leurs applications. L'importance de la sécurité dans le cycle de vie des applications nous amène à nous poser la question suivante :

Comment être capable de livrer de nouvelles fonctionnalités et applications de manière rapide et stable tout en garantissant la sécurité de ce qui est livré ?

Pour répondre à cette problématique, j'ai dans un premier temps présenté l'approche DevOps, son principe de fonctionnement ainsi que l'ensemble des avantages et inconvénients que présentaient cette méthode. Ensuite, je me suis intéressé à la méthodologie DevSecOps et ses enjeux, puis à comment pouvoir la mettre en place. Enfin, j'ai clôturé ce mémoire par un exemple d'application au sein de Dassault Systèmes.

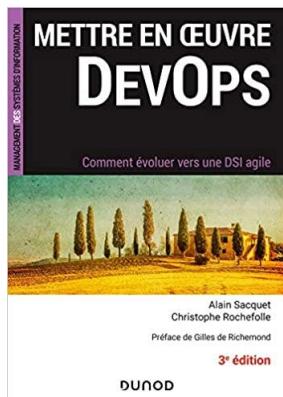
A travers la mise en place d'une approche DevSecOps, avec qui la sécurité devient une composante naturelle du processus de développement, il est plus facile et moins coûteux d'intégrer des mesures de sécurité au cœur du logiciel dès le début pour obtenir à la fois une sécurité améliorée et une satisfaction des clients garantie. L'automatisation de la sécurité dans DevOps est un aspect qui nécessite de nouvelles technologies et outils. Avec la mise en place d'outils de sécurité et des étapes de tests tels que les tests unitaires ou encore de non-régression au sein de la pipeline DevOps, on peut garantir le développement de logiciel « secured by design ».

Selon un récent rapport de Gartner, 80% des entreprises qui ne parviennent pas à adopter une approche moderne de la sécurité devront faire face à une augmentation de leurs coûts d'exploitation et à une diminution de leur capacité de réaction aux attaques d'ici 2023. Il est clair que les entreprises qui ne parviennent pas à suivre le rythme des technologies de sécurité modernes sont à la traîne, surtout dans un contexte où la main d'œuvre est de plus en plus éloignée.

Dans le paysage numérique actuel, qui évolue rapidement, il est crucial que les entreprises s'adaptent au nombre croissant de cyberattaques qui menacent chaque jour de compromettre la sécurité des applications. Les organisations ne peuvent pas se permettre de laisser la sécurité au second plan, c'est pourquoi il est primordial de commencer dès maintenant à intégrer les pratiques DevSecOps dans le développement des applications.

13. Bibliographie

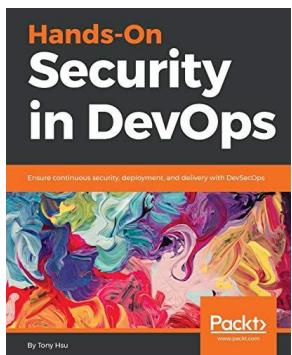
Livres



Titre : Mettre en œuvre DevOps.

Auteurs : Alain Sacquet – Christophe Rochefolle

Description : présentation de la méthodologie DevOps et de recommandations pour la mise en place de cette approche ainsi que du workflow de développement.



Titre : Security in DevOps

Auteur : Tony Hsu

Description : présentation de techniques et d'outils à mettre en œuvre afin d'appliquer l'approche DevSecOps à chaque couche du DevOps. Ensemble de bonnes pratiques de sécurité à appliquer pour améliorer la sécurité de ces applications mais également de son infrastructure, qu'elle soit « *On Premise* » ou dans le Cloud.

Webographie

[1] **Dassault Systèmes.** A propos de Dassault Systèmes [En ligne] (Page consultée le 14/01/2021)

Disponible sur :

<https://www.3ds.com/fr/a-propos-de-3ds/>

[2] **20minutes.** Les attaques informatiques criminelles multipliées par quatre en 2020.

12/01/2021

[En ligne] (Page consultée le 16/01/2021)

Disponible sur :

<https://www.20minutes.fr/high-tech/2950931-20210112-attaques-informatiques-criminelles-multipliees-quatre-2020>

[3] **Le Point.** Cyberattaques : les entreprises françaises touchées plus que jamais.

09/12/2020

[En ligne] (Page consultée le 12/01/2021)

Disponible sur :

https://www.lepoint.fr/high-tech/internet/cyberattaques-les-entreprises-francaises-touchees-plus-que-jamais-09-12-2020-2404883_47.php

[4] **ALM Developpez.** 97% des entreprises informatiques utiliseraient les méthodes de développement agiles. 08/05/2019 [En ligne] (Page consultée le 19/03/2021)

Disponible sur :

<https://alm.developpez.com/actu/260246/97-pourcent-des-entreprises-informatiques-utiliseraient-les-methodes-de-developpement-agiles-selon-un-rapport-de-CollabNet-VersionOne>

[5] **Novencia.** Les chiffres vertigineux de la cybercriminalité. 08/09/2020 [En ligne]

(Page consultée le 19/03/2021)

Disponible sur :

<https://www.novencia.com/chiffres-vertigineux-cybercriminalite>

[6] **NetApp.** Qu'est-ce que le DevOps ? [En ligne] (Page consultée le 16/01/2021)

Disponible sur :

<https://www.netapp.com/fr/devops-solutions/what-is-devops/>

[7] **Atlassian.** Qu'est-ce que DevOps ? [En ligne] (Page consultée le 14/01/2021)

Disponible sur :

<https://www.atlassian.com/fr/devops>

[8] **Journal Du Net.** Le nouvel enjeu pour les développeurs : le DevOps. 30/11/2016 [En ligne]

(Page consultée le 18/01/2021)

Disponible sur :

<https://www.journaldunet.com/solutions/dsi/1189054-le-nouvel-enjeu-pour-les-developpeurs-le-devops/>

[9] **RedHat.** What is Continuous Delivery ? [En ligne] (Page consultée le 18/01/2021)

Disponible sur :

<https://www.redhat.com/fr/topics/devops/what-is-continuous-delivery>

[10] **QRP International.** DevOps c'est quoi ? [En ligne] (Page consultée le 18/07/2021)

Disponible sur :

<https://www.qrpinternational.fr/blog/gestion-des-services-informatiques/devops-cest-quoi-definition-devops/>

[11] **DevOpsSec.** Les avantages du DevOps. [En ligne] (Page consultée le 23/06/2021)

Disponible sur :

<https://devopssec.fr/article/les-avantages-du-devops#begin-article-section>

[12] **New Relic.** What is DevOps ? [En ligne] (Page consultée le 23/06/2021)

Disponible sur :

<https://newrelic.com/fr/devops/what-is-devops>

[13] **MicroFocus.** Ces pratiques de DevOps dans les entreprises de mainframe. [En ligne]

(Page consultée le 08/05/2021)

Disponible sur :

https://www.microfocus.com/fr-fr/media/white-paper/real_world_devops_for_mainframe_enterprises_wp_fr.pdf

[14] **Red Hat.** What is DevSecOps ? [En ligne] (Page consultée le 19/12/2020)

Disponible sur :

<https://www.redhat.com/en/topics/devops/what-is-devsecops>

[15] **Oracle.** Qu'est-ce que l'approche DevSecOps ? [En ligne] (Page consultée le 21/12/2020)

Disponible sur :

<https://www.oracle.com/fr/security/definition-approche-dev-sec-ops.html>

[16] **GeekFlare.** Une introduction à DevSecOps 02/10/2020 [En ligne]

(Page consultée le 04/07/2021)

Disponible sur :

<https://geekflare.com/fr/devsecops-introduction/>

[17] **Journal Du Net.** DevOps ou DevSecOps : pourquoi sont-ils importants aujourd'hui et comment s'intègrent-ils ensemble ? 16/03/2021 [En ligne]

(Page consultée le 04/07/2021)

Disponible sur :

<https://www.journaldunet.com/solutions/dsi/1498661-devops-ou-devsecops-pourquoi-sont-ils-importants-aujourd-hui-et-comment-s-integrent-ils-ensemble/>

[18] **EBRC.** DevSecOps, la sécurité en plus. 02/01/2020 [En ligne] (Page consultée le 04/07/2021)

Disponible sur :

<https://www.ebrc.com/fr/entreprise/blog/devsecops-la-securite-en-plus>

[19] **Ionos.** Qu'est-ce que le DevSecOps et à quoi sert-il ? 11/08/2020 [En ligne]

(Page consultée le 11/01/2021)

Disponible sur :

<https://www.ionos.fr/digitalguide/serveur/securite/quest-ce-que-le-devsecops/>

[20] **Adikts.** Les outils de sécurité applicative 03/02/2021 [En ligne]

(Page consultée le 17/06/2021)

Disponible sur :

<https://www.adikts.io/les-outils-de-securite-applicative/>

[21] **Oracle.** Qu'est-ce qu'un WAF ? [En ligne]

(Page consultée le 19/07/2021)

Disponible sur :

<https://www.oracle.com/fr/security/waf-definition-pare-feu.html>

[22] **CyberArk.** Qu'est-ce que le DevOps ? [En ligne]

(Page consultée le 19/07/2021)

Disponible sur :

<https://www.cyberark.com/fr/what-is/devops-security/>

[23] **EVA Group.** La recette efficace pour mettre en place la démarche DevSecOps

10/05/2021 [En ligne] (Page consultée le 19/07/2021)

Disponible sur :

<https://blog.evagroup.fr/2021/05/10/la-recette-efficace-pour-mettre-en-place-la-demarche-devsecops/>

[24] **La Grotte du Barbu.** Introduction à Hashicorp Vault 23/09/2020 [En ligne]

(Page consultée le 20/07/2021)

Disponible sur :

<https://www.grottedubarbu.fr/introduction-vault-hashicorp/>

[25] **Geekflare.** Une introduction à DevSecOps pour les débutants 02/10/2020 [En ligne]

(Page consultée le 19/07/2021)

Disponible sur :

<https://geekflare.com/fr/devsecops-introduction/>

[26] **By The Way.** Remote Desktop Services [En ligne] (Page consultée le 21/07/2021)

Disponible sur :

<https://www.bytheway.fr/rds-tse/>

[27] **App Dynamics.** What's the difference between DevOps and DevSecOps ? 14/04/2021

[En ligne] (Page consultée le 08/08/2021)

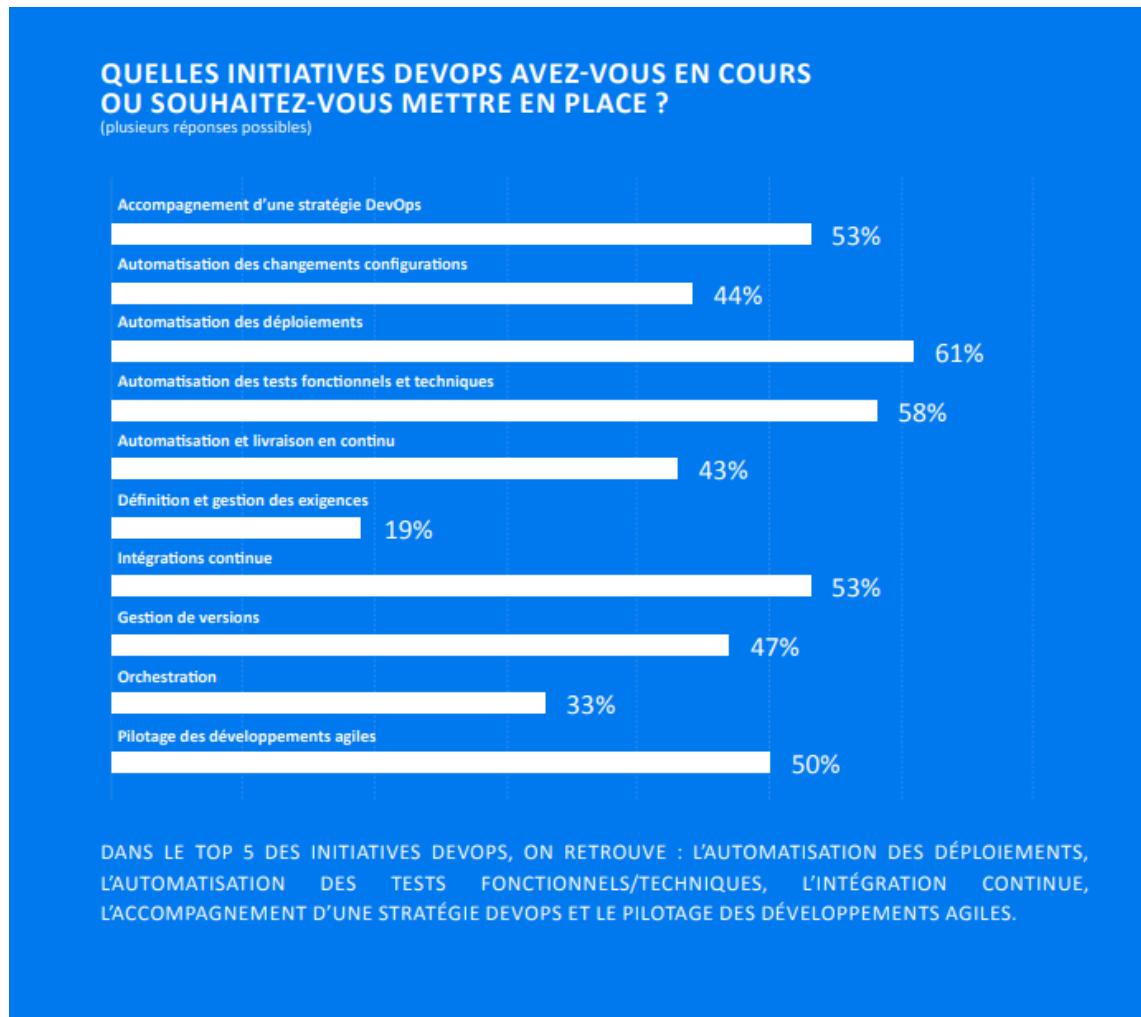
Disponible sur :

<https://www.appdynamics.com/blog/product/devops-vs-devsecops/>

14. Annexes



Annexe 1: Infographie DevOps



DEVOPS ET SÉCURITÉ DANS LE CYCLE APPLICATIF

Dans cette étude, les initiatives d'automatisation du cycle de vie applicatif sont considérées prédominantes pour les répondants. De nombreuses entreprises voient l'automatisation comme un moyen d'accroître la productivité, la rapidité d'exécution, mais également de réduire les tâches répétitives - également sources d'erreurs humaines.

Sur le terrain, nous constatons que nos clients les plus avancés mettent en place des centres de compétences dédiés à l'automatisation des tâches.

Par ailleurs, l'agilité et la transformation numérique semblent également être au centre des débats pour l'adoption de l'automatisation, dû au cycle itératif de développement induit.

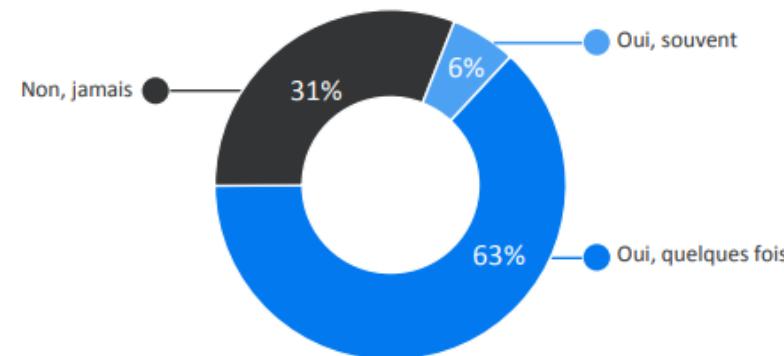
Au-delà des démarches d'automatisation, nos clients insufflent un changement de culture auprès de l'ensemble de la DSI en termes de pratiques de sécurité. On le constate depuis l'avènement de nouvelles législations comme la RGPD, qui contraint les entreprises à gérer plus efficacement la protection et la confidentialité des données de leurs clients.

Parmi les causes explicatives :

- Le manque de sensibilisation de l'organisation IT,
- Les bonnes pratiques de gestion des problèmes sont peu adoptées par les DSI, ce qui n'encourage pas la définition et la mise en oeuvre de plans de prévention,
- Le manque de formalisme des retours d'expérience, ne permettant pas de capitaliser en transverse de la DSI,
- Les boucles de rétroaction des Ops vers les équipes de Dev Agiles trop rares, ne permettant pas d'enrichir en continu le relevé des besoins en terme de sécurité applicative.

Il est également à noter qu'1/3 des répondants estiment n'avoir jamais rencontré d'incidents de sécurité. Cette réponse pourrait traduire le fait que, pour certains répondants, il y a une incapacité partielle de la DSI à détecter les vulnérabilités. Très probablement, ces répondants ne réalisent peu voire pas de tests d'intrusions.

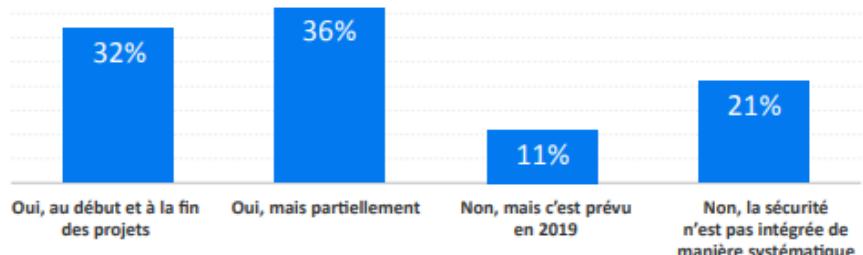
AVEZ-VOUS DÉJÀ RENCONTRÉ DES INCIDENTS DE SÉCURITÉ À LA MISE EN PRODUCTION APPLICATIVE ?



69%

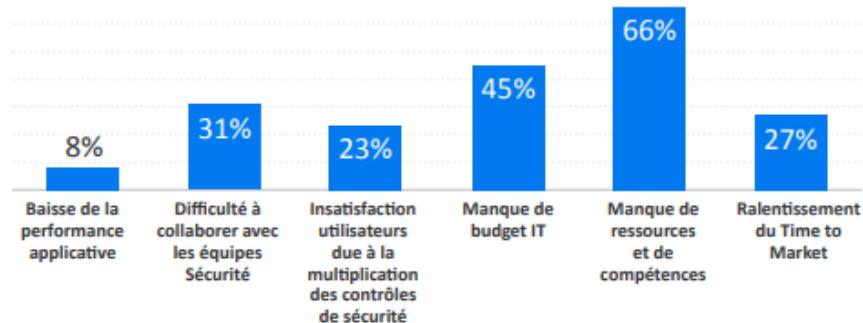
DES RÉPONDANTS ADMETTENT AVOIR DÉJÀ SUBI DES INCIDENTS DE SÉCURITÉ.

INTÉGREZ-VOUS DÉJÀ LA PRÉVENTION DES RISQUES ET LA SÉCURITÉ DANS VOS PROJETS DEVOPS ?



11% DES RÉPONDANTS N'INTÈGENT PAS ENCORE LA SÉCURITÉ LORS DE LEURS DÉVELOPPEMENTS ET 21% PAS SYSTÉMATIQUEMENT.

QUELS SONT LES FREINS À UNE DÉMARCHE 100% DEVSECOPS ?



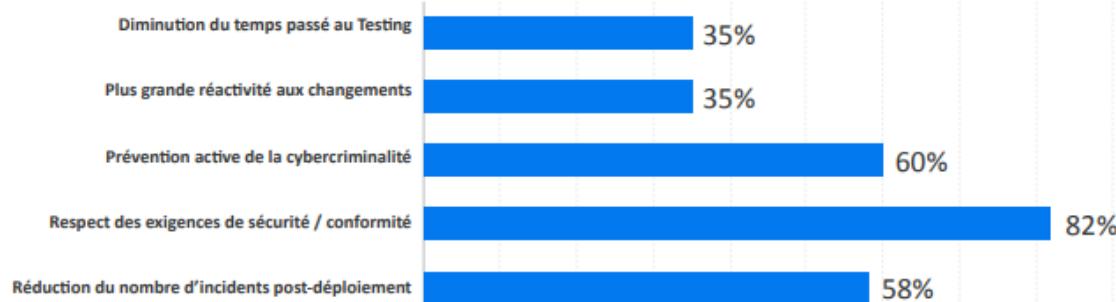
LE MANQUE DE RESSOURCES ET DE COMPÉTENCES RESTE DE LOIN LE PRINCIPAL FREIN (66%) SUIVI DU MANQUE DE BUDGET (45%).

Plusieurs raisons possibles peuvent expliquer ces réponses :

- **Le ralentissement du cycle de livraison continue.** Les équipes de développement peuvent ressentir une certaine frustration concernant le temps nécessaire à la bonne construction d'un code sécurisé. Ce ressenti peut aussi être lié au fait que les équipes de développement sont confrontées à la rigidité de processus imposés par la sécurité.
- **L'exécution manuelle des tests de sécurité,** lorsqu'ils existent. Le manque d'automatisation peut aussi être un élément important.
- **Le manque de temps alloué aux exigences de sécurité.** Il paraît assez évident que les équipes ne consacrent pas le temps nécessaire à la définition des exigences de sécurité, tout comme aux tests permettant d'assurer le respect des exigences par l'application.
- **Une mauvaise priorisation.** Parfois, pour des enjeux de « Time-to-Market », la sécurité n'est pas considérée comme prioritaire dans les organisations.
- **Une absence de considération en amont de la phase de développement.** Enfin, il semblerait que pour 2/3 des répondants, la sécurité n'est pas intégrée dès le début de la phase de développement.
- **Un manque de compétences des équipes.** D'après les répondants de cette enquête, les équipes agiles ne sont pas assez sensibilisées aux bonnes pratiques de sécurité informatique. Ces dernières sous-estiment même leurs compétences en matière de cybersécurité.

QUELS SONT LES AVANTAGES D'UNE DÉMARCHE 100% DEVSECOPS ?

(plusieurs réponses possibles)



LE RESPECT DES EXIGENCES DE SÉCURITÉ ET DE CONFORMITÉ EST L'AVANTAGE LE PLUS CITÉ.

LES RÉPONDANTS PERÇOIVENT TRÈS BIEN LES AVANTAGES D'UNE INITIATIVE DEVSECOPS, MAJORITYALEMENT EN CE QUI CONCERNE LE RESPECT DES EXIGENCES DE SÉCURITÉ ET DE CONFORMITÉ 82%.

En effet, les entreprises qui ne seraient pas conformes aux nouvelles réglementations comme la RGPD se verraient infliger des pénalités pouvant aller jusqu'à 4% de leur chiffre d'affaires annuel. Les premières sanctions en France sont déjà tombées ! Par exemple, la CNIL a infligé une sanction record de 50 M€ à Google pour manquement à ses obligations au règlement général européen de protection des données.

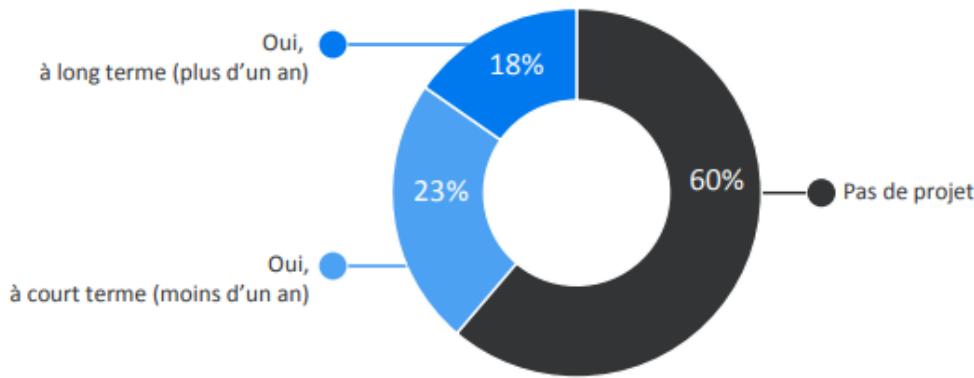
On peut également citer la protection des données clients, préoccupation majeure des DSIs. L'atteinte à l'image de l'entreprise en cas de fuite de données avérée lui est très préjudiciable. **La confiance des clients est très difficile à gagner, mais très facile à perdre.**

L'automatisation permet aussi aux équipes de développeurs de pouvoir revenir à des versions antérieures rapidement en cas de failles de sécurité liées au code.

Enfin, plus un bug est détecté tardivement dans le cycle de vie, plus son coût de résolution est élevé. En effet, un bug de sécurité repéré en production coûte 100 fois plus cher à résoudre que lorsqu'il a été détecté dans la phase de spécification. En automatisant les tests de sécurité sur l'ensemble du cycle de vie applicatif, le coût potentiel de non qualité pour l'entreprise sera infiniment moindre (sans évoquer son image de marque). De plus, le traitement des vulnérabilités de façon intégrée permet de ne pas ralentir le rythme de livraison.

Les initiatives DevSecOps semblent s'installer progressivement dans les entreprises. En effet, près de la moitié des répondants ont déclaré qu'ils allaient investir dans cette voie à court et moyen terme, ce qui atteste d'une nouvelle prise en considération et de sa légitimité pour la mise en conformité.

AVEZ-VOUS UN PROJET D'INVESTISSEMENT DEVSECOPS ?



41%

DES RÉPONDANTS ONT UN PROJET DEVSECOPS,
CE QUI MONTRÉ LE DYNAMISME DE CETTE NOUVELLE TENDANCE.