Software Craftsmanship

Au-delà du code, découvrez les clés d'un développement applicatif réussi

LES CONTRIBUTEURS

Gaël Dupire • Gaëtan Eleouet • Foucault Dupleix • Romain Cheron • Johan Klein

meritis

[Sommaire]

[Bio des auteurs]

Gaël Dupire • Directeur de l'expertise Cloud & Infra et Développeur Senior

Gaël a amorcé son parcours en tant que Consultant en Logiciel avant d'évoluer en tant que Senior Software Development Engineer à la Société Générale (SGCIB), puis d'intégrer l'équipe de Meritis il y a 13 ans.
Titulaire d'un Doctorat en Mathématiques Appliquées de l'UTC, il nourrit une passion ardente pour les mathématiques et leur application concrète. Son amour pour le codage transcende les langages, jonglant habilement avec .Net, Python, C++, et Matlab, démontrant ainsi une polyvalence remarquable dans l'art du développement logiciel.

Gaëtan Eleouet • Directeur de l'expertise Software Engineering

Gaëtan, fort de deux décennies d'expérience en développement professionnel, a consacré 15 années passionnantes au sein de Meritis, où il a évolué en tant que développeur, coach et membre éminent de l'équipe de programmation compétitive.
Son expertise, forgée au fil des années, se concentre désormais sur les applications Java, où il excelle en apportant un soutien remarquable tant sur le plan technique que dans son leadership.
Titulaire d'un diplôme de l'ENSEEIHT, Gaëtan incarne l'esprit d'un créateur dans l'âme, restant toujours proche du code. Sa persévérance et sa passion pour l'innovation ne cessent d'inspirer ceux qui l'entourent.

Foucault Dupleix • Directeur de l'expertise Projets, program & products et Agilité à l'échelle

Foucault, fort d'un Master en Mathématiques, Informatique et Application aux Sciences obtenu à la faculté des sciences de Nantes, a entamé sa carrière en tant que développeur dans le secteur financier. Progressivement, il a endossé des responsabilités de management, de gestion de projet et de Business Analyst dans divers domaines, adoptant une approche de plus en plus agile tout au long de son parcours.
Depuis plusieurs années, il s'est engagé en tant que coach auprès d'entreprises de tous horizons, les guidant avec expertise dans leur transformation et l'adoption de pratiques agiles pour optimiser leur fonctionnement.

Romain Cheron • Responsable de la practice DevOps et Ingénieur DevOps

Romain a acquis une expérience significative en tant que Développeur Java, accumulant plus de 15 années d'expertise avant de se diriger vers le domaine stimulant du DevOps.
Il a intégré l'équipe de Meritis il y a deux ans, apportant avec lui un bagage solide et diversifié.
Titulaire d'un diplôme de Polytech Nice Sophia, son insatiable curiosité l'incite constamment à explorer de nouvelles technologies avec une passion débordante.

Johan Klein • Responsable de la practice Cybersécurité et administrateur système réseau et cloud

Johan a initié son parcours professionnel en tant que Responsable de Centre de Service, évoluant ensuite vers le poste d'Administrateur Systèmes et Cloud chez IBSA Pharma avant de rejoindre l'équipe de Meritis il y a deux ans.
Titulaire d'un diplôme de l'université de Toulon, sa passion se concentre sur la cybersécurité, la protection des infrastructures et des Endpoint. Son engagement s'étend également à l'accompagnement de collaborateurs et clients dans l'adoption des meilleures pratiques en matière de sécurité informatique.

Introduction

Le Software Craftsmanship, également connu sous le nom d'« artisanat du logiciel » ou encore d'« artisanat logiciel », émerge comme une approche de développement logiciel qui prend ses racines d'un livre publié dans les années 90', mais dont le mouvement voit réellement le jour en 2008. Inspiré par la métaphore artisanale, il met l'accent sur les compétences pointues et l'expertise des développeurs. Comme les artisans qui manifestent une fierté inébranlable dans leur travail, le Software Craftsmanship repose sur des valeurs essentielles telles que l'excellence, le pragmatisme et la transmission du savoir.

Toutefois, le craftsmanship doit aujourd'hui s'adapter et composer avec la nouvelle réalité à laquelle sont confrontées les entreprises, tant d'un point de vue technologique qu'humain.

En effet, d'une organisation centrée autour d'un profil expert, le développement applicatif est aujourd'hui passé à une équipe composée de compétences multiples et diverses, tech et métier confondues, amenées à travailler ensemble pour répondre à des besoins de plus en plus complexes. La raison est double : la pandémie de covid-19 et l'intelligence artificielle ont poussé les entreprises à penser différemment leur modèle et à appréhender autrement le développement.

Dans un premier temps, la crise sanitaire a redessiné le périmètre de fonctionnement des entreprises et, par extension, du développement applicatif. L'explosion du télétravail notamment a considérablement transformé l'organisation au quotidien rendant les SI plus exposés.

Parallèlement, l'intelligence artificielle, et plus spécifiquement l'IA générative, a rapidement révolutionné divers usages, ouvrant ainsi de nouvelles perspectives. Que ce soit dans la rédaction de textes, la création visuelle ou la génération de code, les entreprises se trouvent contraintes de s'adapter à cette évolution rapide pour rester compétitives dans cette course effrénée.

Résultat, 60 % des PME victimes d'une cyberattaque déposent le bilan dans les 18 mois suivants[1], témoignant de l'enjeu critique d'adapter son approche du développement applicatif. Pourquoi ? Parce que, au sein d'un environnement en transformation permanente, le développeur doit désormais faire preuve de flexibilité et s'ouvrir davantage aux autres profils de l'entreprise.

Dans cette optique, il est essentiel qu'il fasse partie d'une organisation qui encourage ces interactions. Aujourd'hui, la communication joue un rôle vital.

Chaque individu doit être en mesure de collaborer et d'échanger avec l'équipe de développement à chaque phase du processus pour donner vie à l'application au sein de la chaîne de valeur.

Pour y parvenir, les entreprises doivent alors s'appuyer sur un socle de qualités humaines très fortes. C'est dans leurs soft

skills plus que dans leur appétence technologique que les développeurs trouveront la clé de leur réussite.

Bonne lecture.

[1] Selon une étude de l'assureur Stoïk, juin 2023

[Partie 1]

La relation client comme principe de développement

[Étape #1]

Établir une relation de confiance

**Mener un projet de développement applicatif repose sur le partage de valeurs.
Objectif : faire en sorte que tous les membres de l'équipe soient alignés pour proposer un environnement de développement sain.**

Comment créer une relation de qualité ?

La qualité de la relation client repose sur la capacité à bien comprendre le besoin utilisateur. Objectif : créer un environnement de confiance et faciliter la communication tout au long du projet pour que chacun puisse assumer son rôle.

1. La première étape consiste à se demander quel est le but de l'application et ce que l'on souhaite en obtenir. Cette question est essentielle car elle détermine les profils qui composeront l'équipe de développement.
2. Quels que soient les profils retenus, certaines compétences s'avèrent indispensables, comme la curiosité et surtout l'empathie qui va permettre aux différents profils de travailler ensemble.
3. Le partage de l'expérience est aussi capital : entre développeurs expérimentés et juniors, entre les métiers et les profils tech, etc.
4. Et bien sûr, il faut une certaine appétence pour la technologie car certains développeurs peuvent être amenés à coder au-delà de leur expertise.

Les points d'attention

Comme dans tout projet IT, la communication est primordiale. Plus elle sera précise et claire, moins les risques d'incompréhension seront importants. Attention donc à l'ubiquité du langage : dans le code, certains termes sont tellement génériques qu'ils ne signifient rien (par exemple : helper). Il est donc essentiel de se mettre d'accord en amont sur la signification précise de chaque mot et d'adopter un langage commun pour lever les ambiguïtés et les zones d'inconfort.

Cette communication est également clé pour apporter plus de visibilité sur l'environnement de développement et l'ensemble des profils impliqués. Le chemin est long avant d'être disponible sur le terminal de l'utilisateur final !

Les bonnes pratiques

Difficile de composer avec une multitude de profils et autant de personnalités. C'est pourquoi le point important est de tendre vers un objectif commun, à l'image du chef d'orchestre qui doit mettre en musique une multitude d'instruments. À cette fin, il est crucial d'impliquer les métiers le plus tôt possible dans une optique de co-construction et de co-responsabilité. Le produit final doit être autant métier qu'IT.

[Étape #1]

L'avis de l'expert

Gaël Dupire
Directeur de l'expertise Cloud & Infra et Développeur Senior

« Les clés de succès d'un développement applicatif reposent en amont du projet. Il est fondamental de savoir qui veut créer l'application pour se demander ensuite ce que l'on veut construire, quelles valeurs je veux retrouver, si je préfère aller vite ou avoir une app pérenne… Mais il faut aussi savoir être humble et faire machine arrière lorsque l'on s'est trompé. L'ubiquitous langage* est donc clé car il permet de discuter avec n'importe qui et de déployer un bon esprit critique pour se remettre en question. Cette capacité d'analyse et d'introspection est essentielle dans la vie d'un produit. »

** Le langage ubiquitaire en développement (ou ubiquitous langage en anglais) est un concept qui repose sur la nécessité d'adopter un langage commun à l'ensemble des acteurs impliqués - profils techniques et métier - dans le projet de développement applicatif. Objectif : faciliter la bonne compréhension du modèle de donnée pour pouvoir travailler ensemble et avancer dans le même sens.*

L'échec d'un projet de développement applicatif peut coûter cher à l'entreprise.
En effet, selon le niveau de complexité, le coût d'une application native est estimé entre 30 000€ et 120 000€ ; celui d'une application hybride entre 15 000€ et 80 000€ ; et celui d'une application web progressive (PWA) oscille entre 10 000€ et 50 000€.

Source : Pairform Media Business, septembre 2023

[Étape #2]

Déployer une vue produit

Dans le cadre d'un développement applicatif, il est essentiel d'aborder l'app en tant que produit et non en tant que projet. Pourquoi ? Parce qu'une approche projet privilégie la question budgétaire là où une vue produit permet de prioriser afin de répondre concrètement à un besoin.

Comment s'approcher d'une vue produit ?

Si les principes agiles semblent les plus adaptés au développement applicatif, il importe toutefois de se rappeler que chaque produit et que chaque projet sont uniques et nécessitent une approche sur-mesure pour être efficaces.

1. Le point le plus important repose sur la pertinence et la clarté des besoins. Partir d'une vision très claire permet de donner du sens et d'embarquer les équipes. Le but est de trouver les étapes pour aller vers quelque chose de vertueux sans tout bouleverser. Au risque sinon de se heurter à une véritable réticence au changement.

2. Il est alors possible de définir de petites étapes de la manière la plus logique et optimisée qui soit. L'idée est de faire comprendre aux équipes que l'on n'est jamais dans un mode optimal mais qu'il faut trouver le meilleur mode de fonctionnement à l'instant T, et que celui-ci va évoluer. On parle alors de stratégie produit.

3. Attention néanmoins, cette stratégie produit ne doit pas être déterminée par un manager, un architecte ou un tech lead, mais bien par quelqu'un du métier, proche du terrain, conscient des enjeux et qui maîtrise son sujet.

Les points d'attention

Les développeurs essayent, autant que possible, de privilégier les livraisons fréquentes mais plus petites. Objectif : livrer de la valeur plus régulièrement et prendre aussi moins de risques du fait d'un nombre moins important de changements. Amazon livre ainsi des modifications sur ses applis plusieurs dizaines de fois par jour, et ce de manière transparente. Néanmoins, certains contextes ne permettent pas ce mode de livraison. Par exemple, dans le cadre d'une modification sur le process de fabrication d'un avion, personne ne se risquera à changer ce process toutes les semaines (pour des raisons évidentes de sécurité des voyageurs.

Les bonnes pratiques

L'état d'esprit est essentiel. Tout le monde doit savoir se remettre en question en permanence et collaborer, que ce soit avec les architectes, une autre équipe, le management… Cette collaboration doit s'ancrer dans une optique d'amélioration continue. Enfin, il faut prendre soin de placer les bonnes compétences nécessaires au développement de l'app pour ne pas ralentir le projet.

[Étape #2]

L'avis de l'expert

Foucault Dupleix
Directeur de l'expertise Projets, program & products et Agilité à l'échelle

« Il reste néanmoins très difficile de sortir du spectre projet, surtout en France. Cette approche rassure le management et le métier quant à la maîtrise des coûts, en proposant une date de début et une date de fin. Dans un mode produit, on lance le produit le plus tôt possible, ce qui va nous permettre de vérifier nos hypothèses de départ en termes de valeur, d'attrait de cible, de marché, etc. Si ces analyses sont trop négatives, et que des ajustements ne sont pas suffisants, alors on s'arrête et on n'aura perdu que 3 ou 6 mois. Malheureusement, la culture de l'échec en France rend les projets de développement applicatif compliqués. »

Selon

89%

des responsables, les valeurs à avoir pour une équipe agile performante seraient l'empathie, une culture d'entreprise claire, de bons outils et des dirigeants responsables.

Source : 16e édition du rapport State of Agile, janvier 2023

[Étape #3]

Développer un code propre et de bonne qualité

Le développement logiciel n'est pas qu'une histoire de machine et d'hommes. Il part du besoin client et de sa bonne compréhension par toutes les parties prenantes. Pour y parvenir, la qualité du code est un prérequis.

Comment développer un code de qualité ?

Développer un code propre s'appuie sur la compréhension de son environnement de travail pour savoir où mettre l'accent : sur la performance du code ou sur sa clarté.

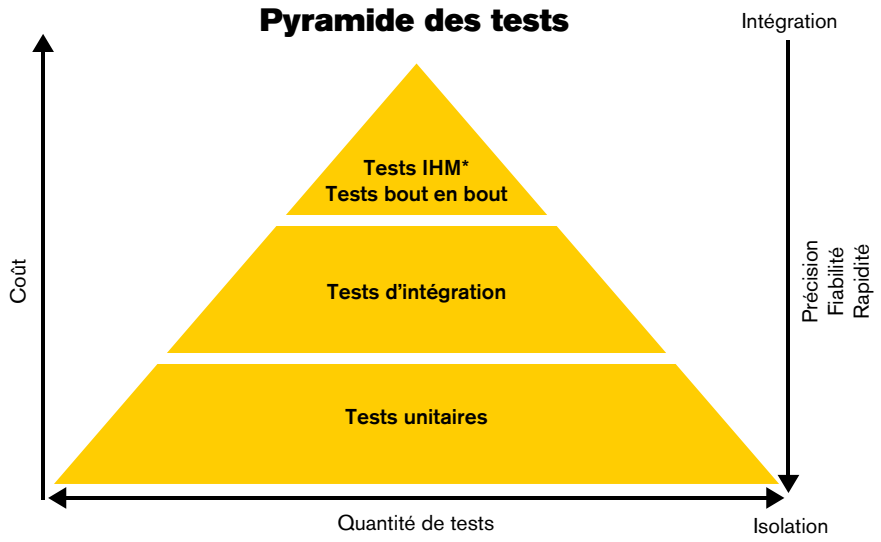
1. Les collaborateurs sont les premiers clients du développeur. D'où la nécessité d'écrire un code qui soit compris par tous, depuis ses collaborateurs plus ou moins experts jusqu'au client final qui veut juste une application simple d'utilisation et qui fonctionne.

2. Voilà pourquoi l'empathie constitue la qualité clé du développement applicatif car elle est la base de construction du langage commun qui facilite la vue produit.

3. En contrepartie, le développeur doit accepter le feedback, si tant est qu'il soit positif et constructif. Il doit donc à la fois être en capacité d'entendre cette critique mais aussi de l'émettre. La remise en cause et l'adaptabilité sont les deux piliers du code propre.

Les points d'attention

Proposer un code de qualité n'est pas sans relever d'une certaine ambiguïté. En effet, on veut souvent faire les choses très vite mais proprement, c'est-à-dire avec un haut niveau de qualité intrinsèque. Une exigence qui s'avère payante sur le long terme mais plus longue à obtenir, ce que le management n'est pas toujours prêt à entendre. Toutefois, ne pas prendre le parti de développer une application qualitative dès le début du projet est toujours un mauvais calcul. Plus tôt les bugs ou problèmes seront détectés, moins ils coûteront à l'entreprise !



** Interface Homme machine*

[Étape #3]

Les bonnes pratiques

Le software craftsmanship est un bon guide concernant l'écriture d'un code qualitatif, un peu à la manière du référentiel ITIL. Le concept : inspiré des principes d'agilité, l'artisanat logiciel a pour but d'améliorer les compétences des développeurs et la qualité technique des projets de développement logiciel à travers trois valeurs phares :

- ➔ La fierté, c'est-à-dire la culture du travail bien fait ;
- ➔ L'excellence, dans une démarche d'amélioration permanente ;
- ➔ Et le pragmatisme afin de prioriser ce qui est réellement important.

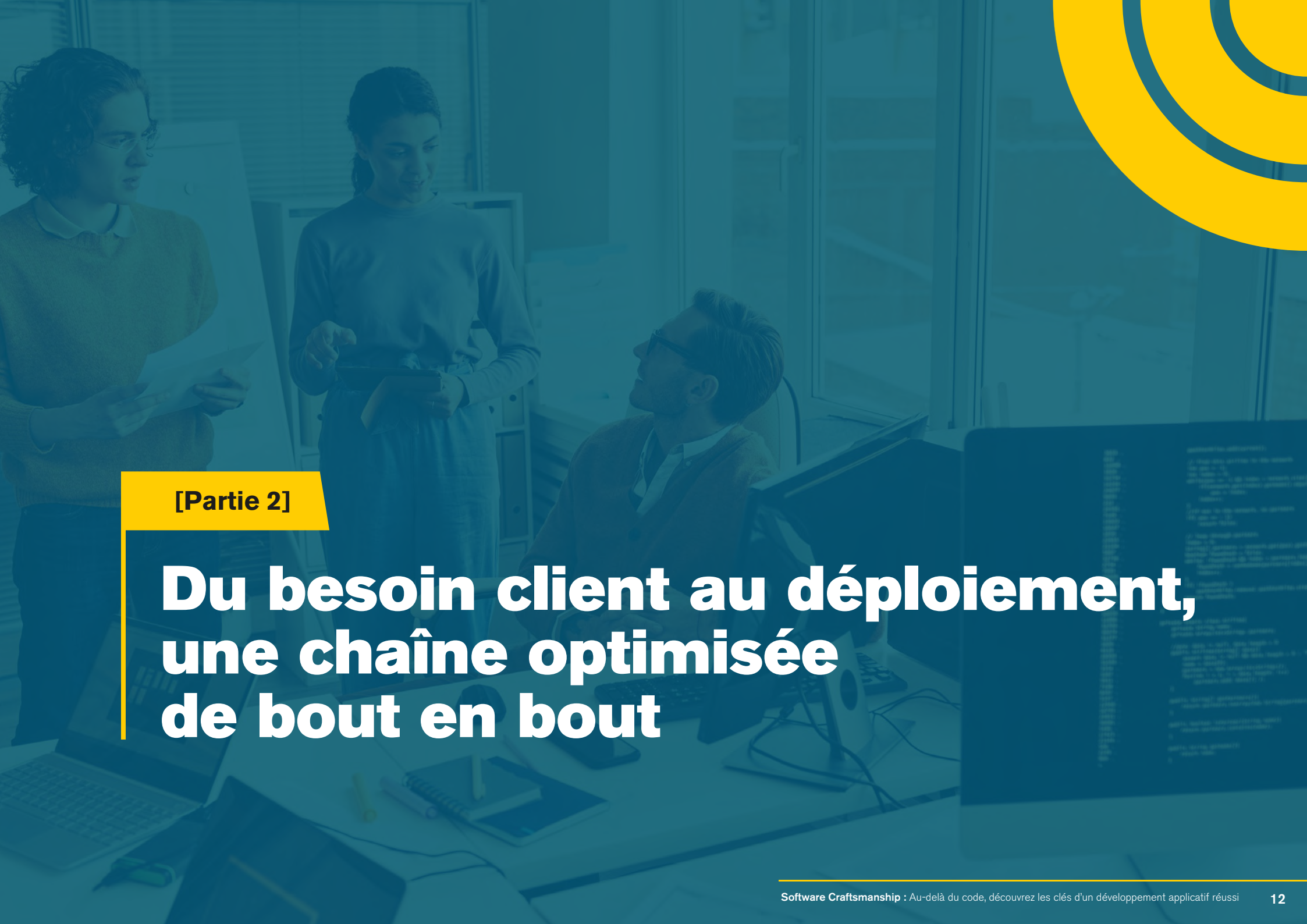
L'avis de l'expert

Johan Klein
Responsable de la practice Cybersécurité

« Établi sur des recommandations plus que sur des normes, le craftsmanship s'alimente du transfert de compétences entre membres, du mentoring, des échanges au sein de l'environnement et de ressources (librairies, applications…) de type Open Web Application Security Project (OWASP), une communauté qui met à disposition notamment un ensemble de recommandations de sécurisation web à suivre, libres et accessibles à tous. D'où l'intérêt de travailler en binôme (Pair Programming) ou à plusieurs (Mob Programming). C'est le principe même de l'amélioration continue. »

Pour en savoir plus

vous pouvez consulter notre article « Agilité et Craftsmanship : quelles interactions ? »



[Partie 2]

Du besoin client au déploiement, une chaîne optimisée de bout en bout

[Étape #4]

S'assurer de l'adéquation du produit aux attentes client

Donner vie à un produit ne sert qu'un seul objectif : la satisfaction des utilisateurs. Pour s'en assurer, les développeurs doivent alors suivre des indicateurs clés tout au long de la chaîne de développement.

Comment vérifier que l'app répond bien aux besoins client ?

Apporter un produit utile et utilisable le plus vite possible, qui soit construit de manière qualitative et évolutive. C'est toute la complexité du développement applicatif. Heureusement, il existe des méthodes efficaces pour s'assurer que l'on est dans la bonne direction.

1. La première méthode consiste à simplement demander au client si l'application lui convient, si la fonctionnalité répond à ses besoins… Une nouvelle fois, il est important de travailler la proximité : il doit « vivre » le produit, le tester, puis prendre en compte ses feedbacks pour adapter ce qui a été fait. C'est le principe même de l'agilité.

2. En btoc, les campagnes d'évaluation de satisfaction (c'est le principe du Net Promoter Score qui permet d'évaluer un service ou de mesurer la satisfaction client) peuvent apporter de tels feedbacks.

3. Une autre approche, certes plus complexe, consiste à injecter des sondes dans l'application pour vérifier les usages qui en sont faits : les temps de connexion, l'utilisation des fonctionnalités, le parcours utilisateur réel, etc. Attention, tout cela doit être fait en respectant les aspects RGPD !

Les points d'attention

En entreprise, l'avis des personnes les mieux payées (« HIPPO » pour Highest-Paid-Person-Opinion) est souvent celui qui compte. Or les mieux payés ne sont pas forcément les plus experts !

Il est également indispensable de casser les silos le plus vite possible pour rendre les équipes plus autonomes.

Enfin, le turnover aussi est un frein, les départs à répétition étant synonymes de perte de connaissances.

Les bonnes pratiques

Il est important de voir le client comme un partenaire. En l'impliquant dans la construction de l'application, le produit sera non seulement plus qualitatif et apporteur de valeur pour les utilisateurs, mais l'accompagnement sera simplifié. C'est pourquoi l'écoute client est une clé de satisfaction finale. Par conséquent, il ne faut pas imposer sa propre façon de faire mais toujours expliquer pourquoi on fait les choses de cette manière.

Et surtout le plus important : l'état d'esprit, la collaboration et l'intelligence collective permettent de soulever des montagnes, mais il ne faut pas pour autant occulter l'importance de disposer d'un minimum de compétences et d'expériences pour réussir à concrétiser.

[Étape #4]

L'avis de l'expert

Foucault Dupleix
Directeur de l'expertise Projets, program & products et Agilité à l'échelle

« Admettons que l'on consulte un panel de personnes pour décider si on refond l'application en bleu ou en rose, et qu'à 80%, le panel répond rose. Il n'est pas rare de voir le choix final se porter sur le bleu parce qu'il a été observé que cette couleur était plus tendance ou parce que le client VIP préfère le bleu… Sauf que l'on déçoit 80% des membres du panel. Par conséquent, ne demandez pas l'avis des personnes si vous n'êtes pas prêt à le prendre en compte. »

L'agilité permet une collaboration accrue pour

69%

des répondants et un meilleur alignement avec les véritables besoins de l'entreprise à

54%

Source : 16e édition du rapport State of Agile, janvier 2023

[Étape #5]

Améliorer en continu grâce à l'approche CI / CD

Une fois le besoin client recueilli, compris et vérifié, le développement applicatif soit s'ancrer dans une démarche d'amélioration continue. Pourquoi ? Pour détecter et corriger les bugs au fil de l'eau, et éviter l'usine à gaz.

Comment mettre en œuvre l'amélioration continue ?

Rendue possible par l'automatisation, l'approche CI/CD ou « intégration continue / déploiement continu » permet d'accélérer les cycles de livraison des fonctionnalités.

1. Pour rendre possible l'amélioration continue, il importe de suivre les grandes étapes du CI / CD :
   - Le build ;
   - Le test ;
   - Le releasing ;
   - Le déploiement ;
   - L'exploitation ;
   - Et la surveillance.

   À noter : Cette approche CI/CD ne s'applique pas à tous les projets applicatifs. La CI/CD s'applique à un projet de plusieurs mois minimum et plus particulièrement pour un produit appelé à vivre dans le temps (incluant de nouvelles fonctionnalités au fil de l'eau). L'investissement qu'elle représente doit en effet être rentabilisé.

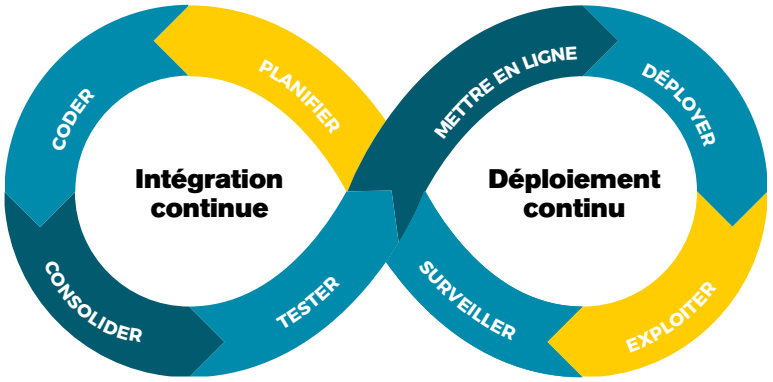
2. La phase de build mérite une attention particulière car elle fournit la première série de retours d'information rapides qui permettent ensuite de lancer les étapes suivantes du pipeline, depuis les tests automatisés jusqu'à la livraison et au déploiement continus. Si cette étape est mal effectuée, il sera alors nécessaire de reprendre toute la chaîne de CI/CD et de refaire les tests.

3. La phase de tests unitaires est elle aussi très importante car elle implique la responsabilité de chaque développeur et une grande rigueur. Quels que soient les outils utilisés, le développeur est toujours responsable du module de code qu'il va implémenter !

Les points d'attention

Certains développeurs sont encore dans une approche fonctionnelle du produit, sans véritable prise de recul. Or déployer une vision globale du produit permet d'anticiper les problèmes techniques, de prioriser les actions et, in fine, de réduire les coûts et le temps de développement. C'est pourquoi l'observabilité est un principe essentiel du développement applicatif. Il consiste à mesurer et à surveiller l'état d'un système avec des métriques issues de « sondes ». Le principe : se focaliser sur les causes des problèmes détectés en analysant en temps réel la façon dont l'applicatif se comporte au sens large dans son écosystème.

Le pipeline CI / CD



[Étape #5]

Les bonnes pratiques

La mesure est la véritable clé du CI/CD. Parmi les KPI les plus pertinents :

- La fréquence de déploiement ;
- Le temps de récupération ;
- Le nombre de bugs identifiés en interne et/ou par les clients ;
- Le nombre de problèmes sur une version par rapport à la précédente ;
- Le nombre d'occurrences d'un même problème ;
- Le nombre d'incidents en production ;
- Etc.

L'avis de l'expert

Gaël Dupire
Directeur de l'expertise Cloud & Infra et Développeur Senior

« Il faut d'abord regarder comment ces KPI sont construits et comment on peut les utiliser. Il est donc nécessaire de connaître les limites de son système et de comprendre ce que l'on est en train de mesurer. Mais plus que des KPI, il faut se fixer des objectifs : que veut-on atteindre ? Il faut aussi challenger les KPI, les modifier et les ajuster dans le cadre de l'amélioration continue. Tout doit être correctement fait dans la chaîne pour qu'elle ait de la valeur. Au-delà, il faut inventer ses propres KPI. On est dans un projet qui s'auto-construit et s'auto-alimente. »

L'implémentation des pipelines de CI / CD représente la

2e raison

qui pousse les entreprises à adopter une plateforme d'ingénierie (20 %). La première reste la productivité des développeurs (21 %).

Source : Platform Engineering in 2023:Rapid Adoption and Impact, CloudBees, 2023

[Étape #6]

Adopter une démarche DevOps

Le DevOps n'est pas une norme ni un recueil de bonnes pratiques. C'est avant tout un modèle centré sur la capacité à faire travailler ensemble des profils totalement opposés. D'où l'intérêt de l'appliquer au développement applicatif.

Comment mettre en œuvre le DevOps ?

Un des principes du DevOps est d'abolir les barrières entre les équipes de développement et de production. In fine, la démarche sert un objectif principal : délivrer plus vite de nouvelles fonctionnalités en production sans pour autant sacrifier la qualité.

1. Si l'automatisation permet de produire plus vite et mieux, elle ne remplace pas les contrôles humains, toujours nécessaires pour assurer la maintenance des outils ou valider la qualité. Le DevOps ou la technologie en général ne remplace pas l'humain qui reste au centre du développement applicatif.

2. Toutefois, pour appliquer cette démarche, des technologies sont nécessaires. C'est là qu'intervient l'automatisation sur des tâches répétitives et chronophages. Sur les plateformes de CI/CD, l'automatisation va ainsi faciliter l'intégration des développements de façon continue et le lancement des tests pour valider les développements, et éviter l'effet tunnel.

3. L'intérêt réel du DevOps est simplement de permettre de satisfaire le client final en livrant rapidement une fonctionnalité de qualité.

Les points d'attention

Le DevOps implique une communication en continu. Ainsi, chez Meritis Group, l'une des questions que pose souvent Gaël Dupire en entretien est la suivante : « Vous développez une app qui produit des logs pour savoir ce qu'il se passe. Vous mettez l'app sur le serveur et la laissez tourner pendant 2 mois. Mais le disque dur plante car il a atteint sa capacité de stockage maximale à cause de ces logs. Qui est donc responsable ? L'Ops qui n'a pas dit que le disque était plein, ou le Dév qui n'a pas pensé à faire du code propre et à nettoyer ses logs ensuite ? » Voilà typiquement un point d'attention à prendre en compte.

Les bonnes pratiques

Comme évoqué précédemment dans ce livre blanc, l'écoute et la prise en compte des feedbacks des utilisateurs métier est essentielle pour accélérer les processus. Elle permet notamment d'intégrer ce qu'il est impossible de mesurer par des indicateurs.

Mais le plus important reste la volonté de collaborer. Un vrai état d'esprit agile, de collaboration et d'intelligence collective lève tous les freins et permet de s'assurer que tout le monde – management, métier et IT – avance dans le même sens.

[Étape #6]

L'avis de l'expert

Romain Cheron
Responsable de la practice DevOps et Ingénieur DevOps

« On voit toujours le monde par son propre prisme. Dans l'informatique de gestion et des systèmes d'information de façon générale, le DevOps est un véritable plus. En revanche, il n'est pas adapté à tous les types de projet. En effet, la livraison itérative ne correspond pas à certains projets qui nécessitent une seule livraison finale. Le DevOps ne va pas résoudre tous les problèmes. Et comme pour le cloud, il peut y avoir des déconvenues lorsque l'approche est mal comprise ou mise en œuvre de façon incorrecte. »

Pour tout savoir du DevOps, téléchargez notre livre blanc

6 facteurs clés pour réussir sa démarche DevOps

[Partie 3]

Le *security by design*, nouveau standard du développement applicatif

[Étape #7]

Évoluer du DevOps au DevSecOps

De la même manière que l'on se demande si l'application répond au besoin client, il faut se poser la question de l'impact de ce besoin exprimé sur la sécurité de l'app et sur les données utilisateurs.

Comment mettre en œuvre le DevSecOps ?

Face à l'explosion des cybermenaces, la sécurité est un besoin prioritaire mais qui n'est pas toujours exprimé. Les utilisateurs ont besoin d'être rassurés par leur environnement. Les développeurs doivent donc se questionner quant à l'infrastructure à mettre en place et adapter le design à la sécurité.

1. Il faut arrêter de sacrifier la sécurité pour des raisons économiques. Le manque de sécurité en a un coût bien plus important. Plus la sécurité sera intégrée en amont, plus les données utilisateurs et de l'entreprise seront protégées.

2. La sécurité est une responsabilité partagée au sein de l'équipe de développement qui doit organisée de manière à prendre en compte les éléments de sécurité dans ses choix de conception aussi bien au niveau architecture, infrastructure et application.

3. La partie sécurité est souvent testée comme une brique en fin de chaîne, une fois la fonctionnalité livrée. Le DevSecOps consiste donc à réfléchir à la sécurité des applications et de l'infrastructure dès le départ : où vont être mes conteneurs, où je vais mettre mon SOC, quelles identités vont se connecter à l'app, comment centraliser les contrôles d'accès, etc.

Les points d'attention

Les entreprises, et plus particulièrement les PME et ETI, sont très exposées en termes de sécurité de leurs informations. Parmi les principales menaces auxquelles elles sont exposées :

- ➔ L'injection de code SQL externe dans le système d'information, ouvrant ainsi l'accès aux bases de données de l'entreprise aux pirates informatiques.
- ➔ Les problématiques autour des authentifications et autorisations : seules les personnes autorisées doivent avoir le droit d'accéder à la donnée !
- ➔ La gestion des dépendances générales : elle implique de disposer d'une équipe capable de comprendre, de surveiller et de corriger le plus vite possible avant qu'une faille ne soit exploitée.

Les bonnes pratiques

L'application du RGPD permet de prendre en compte un certain nombre de bonnes pratiques de sécurité comme la journalisation des événements. Passer par l'écrit peut aussi rendre la sécurité plus transparente.

Mais un regard humain sera toujours plus efficace qu'un outil automatique. Donc il importe de se tenir au courant des bonnes pratiques de développement pour valider l'ensemble, de partager entre elles les ressources et de ne pas hésiter à faire appel à des aides extérieures pour tester ses apps… Plus on augmente le nombre de regards, plus on aura la couverture la plus complète.

[Étape #7]

L'avis de l'expert

Gaëtan Eleouet
Directeur
de l'expertise
Software
Engineering

« Le problème est que personne dans l'équipe ne sait vraiment comment faire tant il y a de métiers qui interviennent sur la chaîne de développement applicatif : développement, mise en production, suivi, gestion de l'infrastructure, donneur d'ordre, client final… Soit autant de profils différents à qui expliquer le besoin et les contraintes de sécurité. Or quand un problème survient, il est souvent déjà trop tard. C'est pourquoi il faut y penser dès le départ et être responsable de ce que l'on fait, à tous les niveaux. »

Dans

60%

des cas, les attaques affectent fortement le business des entreprises, avec pour effets principaux de perturber significativement la production ou de diminuer le chiffre d'affaires.

Source : 8e édition du baromètre annuel du CESIN, janvier 2023

[Étape #8]

Penser au WAAP et au ZTNA

Pour compléter le DevSecOps, il existe certaines mesures et approches spécifiques qui permettent de couvrir un domaine particulier, à l'image des API, ou de proposer au contraire une couverture globale.

Le WAAP

Les API ou interfaces de programmation d'application représentent un risque sous-estimé par la majorité des entreprises. Le WAAP – pour Web Application and API Protection – apporte alors une solution efficace pour les sécuriser.
En agissant comme un pare-feu d'applications web (WAF ou Web Application Firewall), le WAAP permet de centraliser la protection des applications web contre les vulnérabilités et les attaques courantes.

95% **des entreprises ont connu une attaque au cours des douze derniers mois. Face à ces risques :**

- **Plus d'un tiers n'ont aucune stratégie de sécurité des API déployée.**
- **27 % une stratégie basique.**
- **29 % intermédiaire avec quelques tests.**
- **11 % seulement ont mis en place des tests récurrents et une protection dédiée aux API.**

Source : Étude « State of API Security », Salt Mobile, Q1 2023

Les bonnes pratiques

Les API touchent les services en arrière-plan comme les bases de données, l'IAM (Identity Access Management) ou encore l'infrastructure interne. C'est pourquoi ils doivent être sécurisés au maximum car ils ouvrent la porte aux données de l'entreprise. Voici les bonnes pratiques à suivre :

- ➔ Utiliser des protocole SSL (HTTPS) et l'application de TLS 1.2 ou 1.3 (TRansport Security Layer). Les anciennes versions de TLS doivent être dépréciées et le « HTTPS non sécurisé supprimé.
- ➔ Inventoriser les API.
- ➔ Contrôler leurs accès en utilisant par exemple Oauth2 ou OpenID Connect.
- ➔ Appliquer le principe de ZTNA et propager l'identité pour permettre à chaque couche d'un réseau sur lequel résident les services consommés de prendre ses propres décisions pour valider ou non un accès requêté.
- ➔ Mettre en place le WAAP.
- ➔ Tester la sécurisation des API.
- ➔ Surveiller les API (déverser les logs dans un SIEM ou Système de gestion des informations et des événements de sécurité. Ces données peuvent être analysées par un SOC ou Centre Opérationnel de Sécurité en live ou post-attaque).

Pour connaître la liste des points à protéger en priorité, consultez le Top 10 des risques de sécurité concernant les API sur le site de l'OWASP.

Le ZTNA

Le Zero Trust Network Access (ZTNA) est le nouveau standard de sécurité des réseaux. Il limite strictement l'accès des utilisateurs sur la base d'une politique d'autorisation dynamique. Tous les utilisateurs dans l'organisation ou en dehors doivent être authentifiés, autorisés et validés en continu en fonction de la politique ZTNA avant de pouvoir accéder aux applications et aux données. Le principe est le suivant : on ne peut faire confiance à rien de ce qui est connecté au réseau !

Les bonnes pratiques

Le ZTNA consiste à partir du principe qu'on est déjà attaqué pour adopter une vérification permanente impliquant la mise en place d'un certain nombre de règles dynamiques. Objectif : agir comme si votre réseau était victime d'une faille pour limiter l'accès aux ressources, vérifier le chiffrement de bout en bout et surveiller l'activité du réseau. L'idée ici est de se dire que toute source de données et tout service sont considérés comme des ressources ou des identités, que ce soit une API, un ordinateur, un compte, un objet connecté… et par conséquent un point d'entrée vers le réseau. Les communications doivent alors être sécurisées entre chaque objet quel que soit le point d'entrée au réseau.

L'accès est accordé individuellement et par session avec le moindre privilège. En d'autres termes, l'utilisateur ne peut accéder qu'aux ressources dont il a besoin à un moment donné, et uniquement, aussi longtemps qu'il en a besoin lors de cette session. L'accès à la ressource ne permet en effet pas un rebond automatique. Résultat, sur chaque couche du réseau, il est alors demandé de reprouver son identité.

En parallèle, la vérification explicite rend obligatoire l'authentification et l'autorisation de chaque utilisateur et de chaque terminal sur chaque session en utilisant autant de données que possible pour déterminer leur niveau de risque.

L'avis de l'expert

Johan Klein
Responsable de la practice Cybersécurité

« Ces concepts peuvent vivre ensemble. Le WAAP va créer un firewall, centraliser et créer une barrière devant mes API inventoriés. Ensuite, le ZTNA permet d'intégrer la sécurité progressivement. Toutefois, le ZTNA a un coût donc il faut bien réfléchir en amont. Tout dépend de la mobilité des collaborateurs et de la fréquence avec laquelle ils sont susceptibles de se connecter en dehors des murs de l'entreprise. Il est indispensable de pouvoir bloquer les connexions suspectes immédiatement car les entreprises mettent en moyenne 19 jours pour détecter une attaque. En ce sens, la vigie permanente apportée par le ZTNA est une clé de sécurité essentielle pour les entreprises. »

Conclusion

Le développement applicatif va devenir de plus en plus complexe du fait de la multiplication des interactions avec une grande diversité de systèmes. Auparavant autonomes et indépendantes, les applications discutent aujourd'hui avec le monde entier, augmentant drastiquement le nombre de connexions. En parallèle, les équipes doivent composer avec une multitude d'outils en interne affectant la compréhension de leur écosystème.

Et la tendance n'est pas près de s'inverser : l'heure est au développement d'applications plus petites dédiées à un besoin précis plutôt qu'à des solutions plus importantes capables d'adresser l'ensemble des besoins. D'où le phénomène de plateformisation qui permet aux équipes IT de gérer cette masse croissante d'outils.

Mais cette complexité n'est pas sans impact sur la cybersécurité. En effet, si les entreprises ont investi dans des ressources technologiques et humaines, les hackers aussi se sont organisés pour être toujours plus imaginatifs et performants. Par conséquent, la sophistication des cyberattaques rend toujours plus difficile la protection des données sensibles, notamment pour les PME et les ETI.

Dans ce contexte, l'agilité apparaît aujourd'hui inéluctable et s'impose comme la clé du développement applicatif dans les années à venir, à l'image du DevSecOps dont 96 % des « utilisateurs » affirment qu'il est avantageux pour leur entreprise*. Pour autant, aucune méthodologie, ni aucune technologie ne pourra être efficace sans des compétences humaines pour les appliquer, les surveiller et les valider. En effet, quelle que soit la maturité technologique de l'entreprise et l'expertise des développeurs, les qualités et les valeurs humaines sont les véritables assets qui font la différence.

Voilà pourquoi, au-delà de l'appétence à la technologie, l'empathie, la curiosité, la capacité d'analyse et la construction de la critique positive sont sans conteste les clés de succès du développement applicatif.

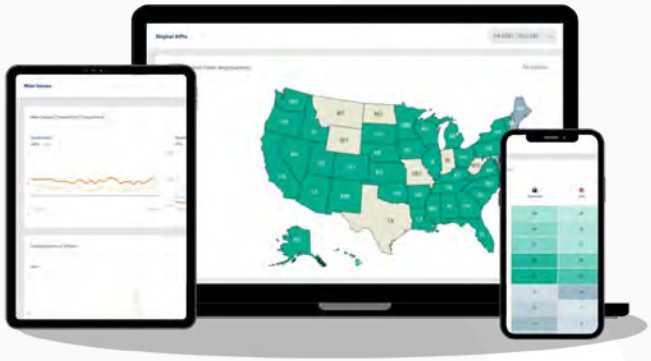
* Selon un rapport d'Infosec, janvier 2023

Nos dernières publications

Cliquez sur l'image pour les découvrir

Cas client

Exploiter et valoriser un grand nombre de données



Télécharger

Retrouvez l'ensemble de nos publications ici

ou sur notre site
https://meritis.fr/livres-blancs/

meritis

Meritis, le talent d'aller plus loin.

CONSEIL, PILOTAGE ET DÉVELOPPEMENT IT

Meritis est une société de conseil en transformation des Systèmes d'Information et Organisations.

➔ **Notre approche ?**
Accompagner nos clients sur l'ensemble de la chaîne de valeur : cadrage personnalisé, pilotage & développement applicatif pour les projets IT.

➔ **Notre mission ?**
Connecter les meilleurs talents au service de la transformation numérique pour donner un temps d'avance aux entreprises.

Nos 900+ consultants vous accompagnent avec agilité dans tous vos projets de transformation digitale.
Un seul objectif : vous emmener plus loin.

Notre expertise Software

Les logiciels existants ne répondent plus à vos besoins spécifiques ? Vous cherchez à les personnaliser ou à les intégrer plus étroitement à votre infrastructure ?

Aujourd'hui, les impératifs technologiques exigent bien plus qu'une simple fonctionnalité. Ils requièrent des solutions logicielles de haute qualité, robustes, sécurisées et alignées sur les objectifs stratégiques de votre entreprise. Chez Meritis, nous saisissons ces enjeux et nous nous engageons à vous accompagner à chaque étape de votre projet.

Nous vous offrons l'opportunité de créer un avantage concurrentiel solide grâce à des applications sur mesure, développées avec une expertise inégalée, répondant précisément à vos besoins spécifiques.

Pourquoi choisir Meritis pour vos solutions logicielles sur mesure ?

➔ **Expertise Technique de Haut Niveau :** Notre équipe pluridisciplinaire, composée de développeurs full stack, d'experts DevOps, de designers UX/UI et d'ingénieurs en données, assure une approche intégrée pour relever vos défis logiciels les plus complexes.

➔ **Adaptabilité Environnementale :** Nous vous accompagnons dans l'adaptation de votre solution logicielle, qu'il s'agisse d'une infrastructure on-premise ou cloud, pour satisfaire pleinement vos exigences.

➔ **Sécurité et Conformité :** Avec notre certification ISO 27001, nous garantissons la sécurité et la conformité de nos solutions pour protéger intégralement vos données sensibles.

CONTACTEZ-NOUS

Nous contacter

Un projet, une question, vous souhaitez en savoir plus ?

Contactez-nous ! Nos équipes d'experts sont votre disposition pour répondre toutes vos questions.

NOUS CONTACTER

NOUS REJOINDRE

meritis.fr

meritis