

# La livraison continue : enfin y voir clair !

Atelier thématique à destination du Pôle Ministériel

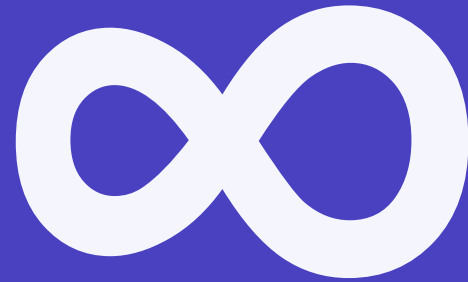
**Proposé et animé par la DNUM**

- **1. Introduction : la livraison continue, bien plus qu'un sujet technique**
- **2. Une pratique d'équipe avant tout**
  - Méthodologie
  - Rituels
  - Feedback
  - Témoignage
- **3. Pour passer à l'action**
  - Livraison continue
  - Les offres DAM

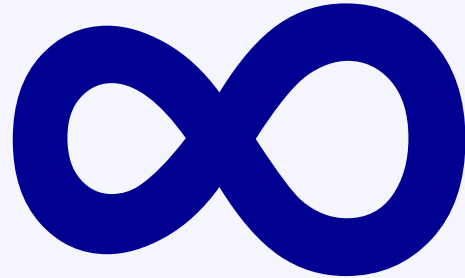
# 1. Introduction : la livraison continue, bien plus qu'un sujet technique !

---

Qu'est-ce que ce symbole vous évoque ?



**Ce symbole “infini” exprime généralement la continuité**



qui s’exprime par la démarche produit (amélioration continue), les méthodes agiles et la **livraison continue**.

# Pour vous, c'est quoi la livraison continue ?



**Si on devait en donner une définition :  
(Wikipédia parce qu'elle est bien)**

*Pratique dans laquelle les équipes produisent des logiciels dans des cycles courts, ce qui permet de le mettre à disposition à n'importe quel moment.*

**En deux mots (ou presque) : Livrer de la valeur souvent.**

**Pourquoi cette pratique est importante ?**

**Et pourquoi ça nous concerne tous, en tant que personnes qui  
conçoivent des logiciels ?**

**Qu'est-ce que ça change vraiment ?**



# 2 exemples pour saisir le concept livraison continue

## **Exemple 1**

*Imaginez que vous construisez la maison de vos rêves. Vous faites les plans.  
Mais vous n'êtes pas autorisé à la voir durant le chantier.*

*Impossible de passer sur le chantier,  
de repérer les erreurs,  
de prioriser les tâches,  
de guider les ouvriers,  
d'échanger avec le chef de travaux*

*De vous confronter au résultat, même si il n'est pas définitif.*

Vous ne voyez le résultat que quand votre maison est **terminée**.  
Vous imaginez l'enjeu et la pression à la livraison de votre maison ?

**À la fin, il y a de fortes chances que vous fassiez une tête qui ressemble à ...**

À la fin, il y a de fortes chances que vous fassiez une tête qui ressemble à ...





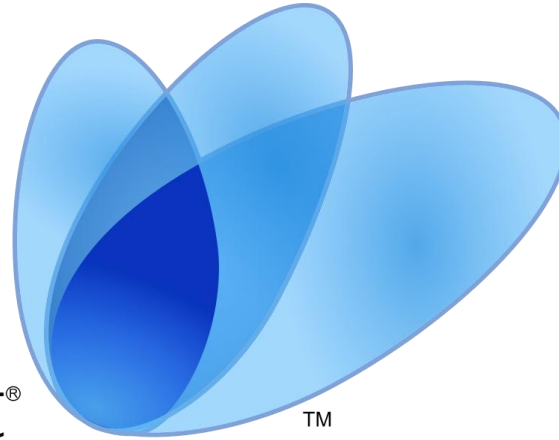
La livraison continue, c'est se confronter sans attendre à ce que l'on a fait, pour ne pas prendre le risque de décevoir nos utilisateurs.

## *Exemple 2*

*Imaginez que vous achetez une encyclopédie numérique mais qui n'évolue jamais.*

*Vous souhaitez accéder à des connaissances actualisées ?  
Il faut racheter une version ! Un peu comme un livre.*

*Cela vous semble une idée complètement absurde ?*



Microsoft®  
Encarta®

*C'était le modèle d'Encarta !*

**Si on devait grossir le trait...**



## Le mise en production cycle en V

Les évolutions sont planifiées,

Lors de la livraison, le périmètre de ces évolutions reprend souvent plusieurs mois de travail

Beaucoup de changements qui impliquent un gros travail de recette et beaucoup de nouveautés pour les utilisateurs

En attendant parfois une nouvelle livraison des mois plus tard

## Vers la pratique de la livraison continue

Les évolutions font partie du quotidien de l'équipe

Lors de la livraison, le périmètre de ces évolutions reprend souvent quelques jours de travail

De petits changements sont proposés, plus faciles à recetter et testables rapidement

Corrigés dans la livraison suivante quelques jours ou semaines après.

À vos questions

---

## 2. Une pratique d'équipe avant tout

---

# Les relations avec l'extérieur

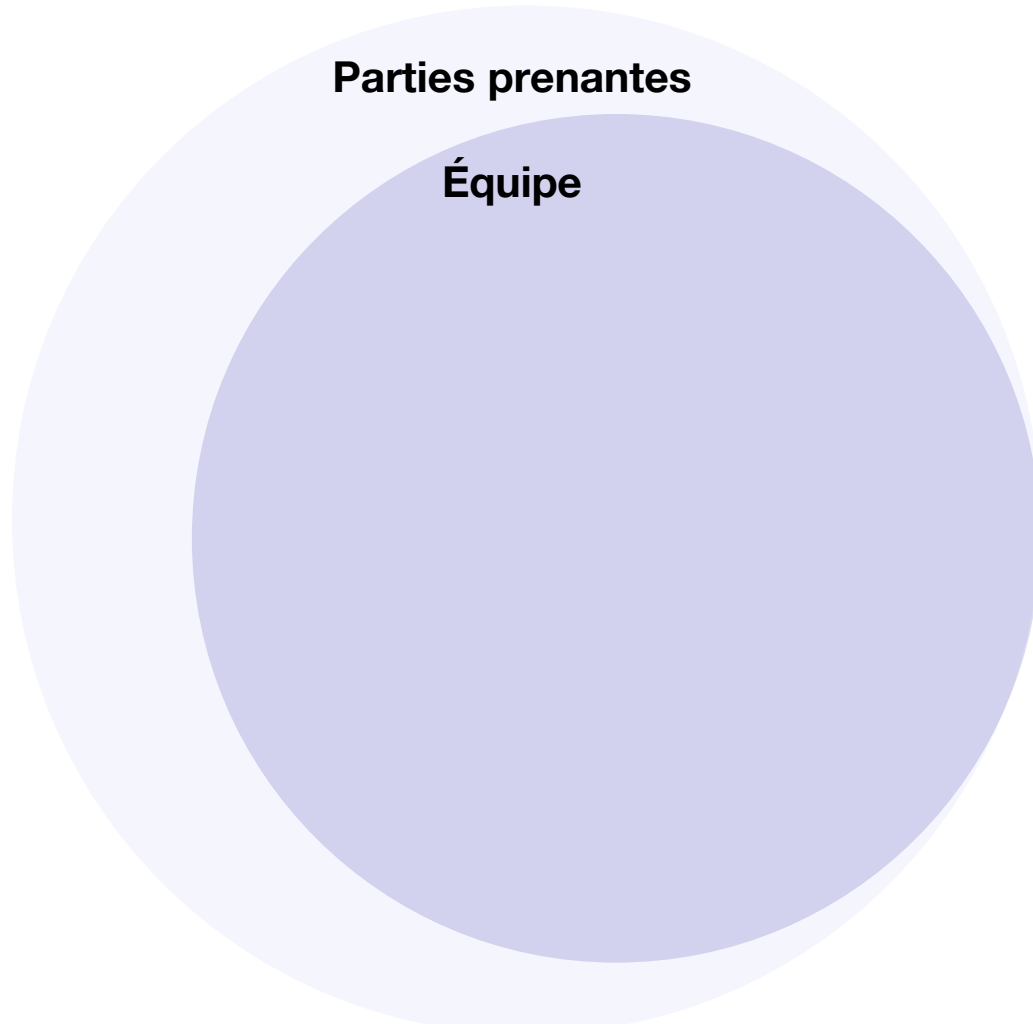
## Parties prenantes

- Rencontrer ses **utilisateurs** en continu : tests et retours réguliers à la fréquence des mises en production
- Rencontrer et collaborer avec ses **sponsors** : réévaluer son budget, sa roadmap régulièrement

### 💡 Pour aller plus loin

- Fiche récap : [Les tests utilisateurs](#) (bonus : [Le Jour où... "J'ai fait mon premier test utilisateur"](#))
- Le [comité de suivi](#) avec ses sponsors

# La composition de l'équipe

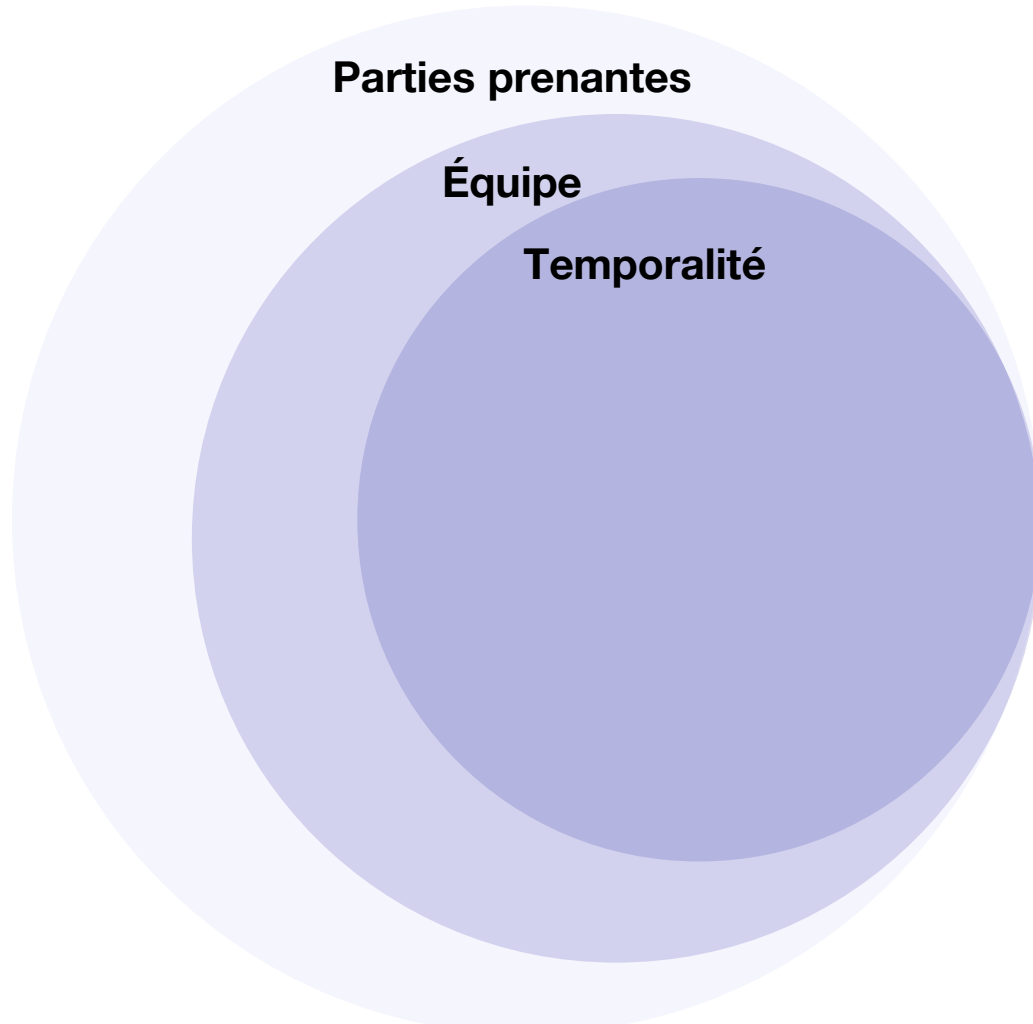


- Une équipe **pluridisciplinaire collaborative** : MOA + MOE + prestataire ensemble, le cahier des charges est remplacé par la présence du MOA en continu !
- Une **disponibilité hebdomadaire** : pour absorber les retours, mises à jours, recette, etc.

## 💡 Pour aller plus loin

- Les [rôles](#) au sein d'une équipe produit PNM
- Fiche récap : [La communication asynchrone](#) en équipe !

# La temporalité de livraison

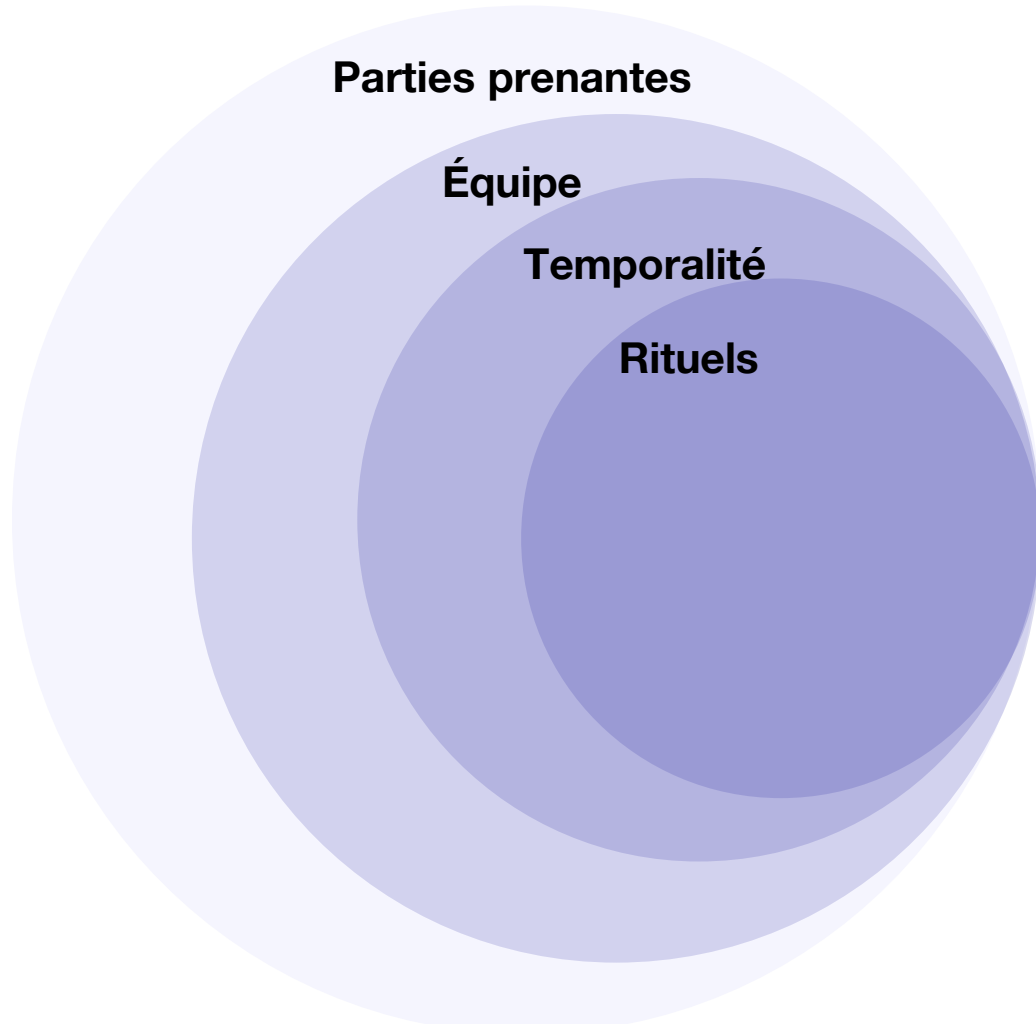


- Définir un **rythme de livraison** adapté : toutes les x semaines
- **Livrer à l'heure** : réduire le scope de livraison si besoin, mais pas de décalage de la date de mep
- Nouveau produit, penser **PMV\*** : un petit scope, mais fonctionnel immédiatement

## 💡 Pour aller plus loin

- \*Produit Minimum Viable
- [SCRUM](#), version résumée une métope qui temporalise
- [Lean Startup](#), une méthode d'amélioration continue

# Les rituels associés



## Chaque semaine

- Conception : mettre à jour son backlog opérationnel, intégrer les retours utilisateurs

## Chaque livraison

- Recette : tester le produit
- Rétrospective
- Planification : prochaine itération

## Chaque trimestre

- Mettre à jour sa roadmap

## Chaque semestre

- Rencontrer ses sponsors

## Chaque année

- Se réaligner sur la vision

### 💡 Pour aller plus loin

- Fiche récap : [Les rituels produit](#)



Voir quel  
angle  
aborder

**Voir quel angle  
aborder**

- Quelle est votre  
orga / rythme de  
livraison ?

*Témoignage dev*

**Christophe Besnard**

*Développeur Fullstack Résorption Bidonvilles*





À vos questions

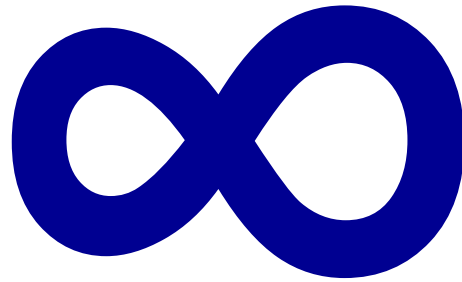
---

## 3. Pour passer à l'action

---

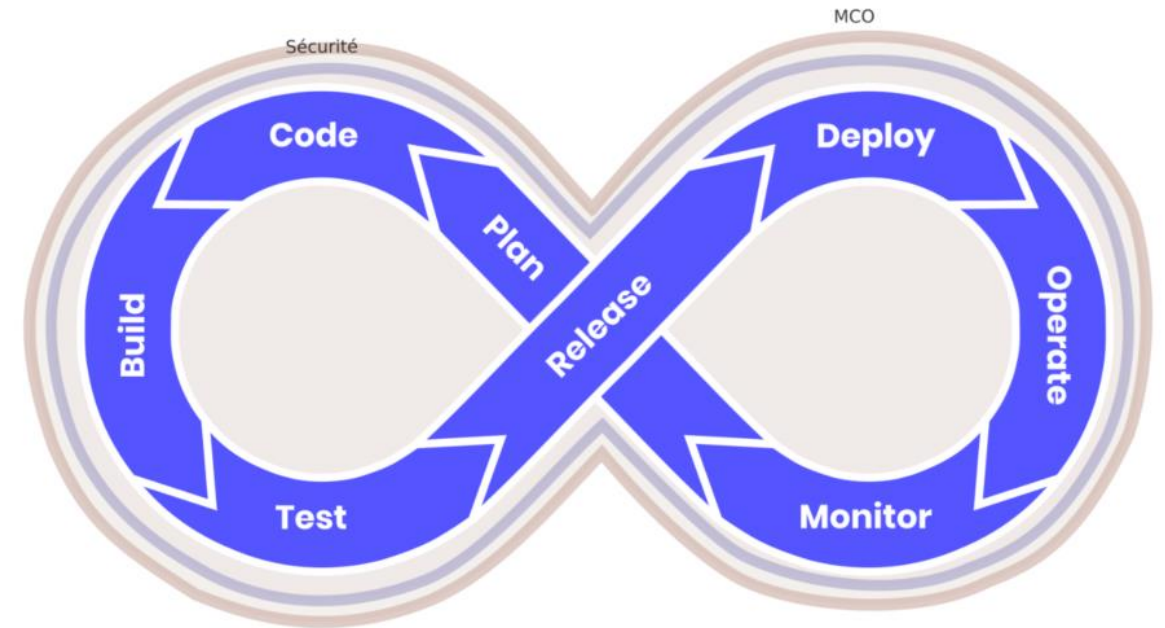
## Objectif de la livraison continue

La livraison continue permet de réduire durablement le délai entre un nouveau besoin validé et le moment où il est déployé : en production, mais possiblement sur d'autres environnements (recette, pré-prod ...)

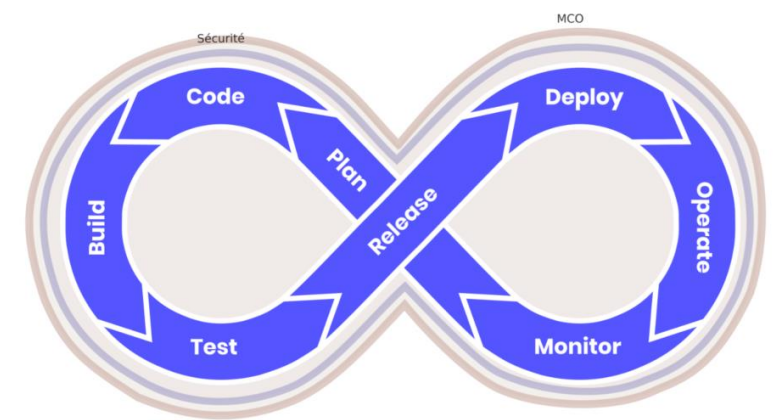


# Principe de la livraison continue

- L'amélioration continue dans les gènes
- Huit étapes structurent l'activité
- L'intégration continue et le déploiement continu en support (automatisation)
- La sécurité et le MCO en fond



# Mettre en oeuvre



Plan : collecte des besoins et récits utilisateurs, priorisation, planning, feuille de route (organisation et socle technique).

Code : réalisation du code, applicatifs, tests, infrastructure codée, configuration, données.

Build : une fois le code mis à jour, phase de construction d'une nouvelle version, automatisée dans un environnement cible.

Test : environnement de tests pour tests automatisés fonctionnels, techniques, sécurité.

Release (livraison) : la version construite du produit est susceptible de passer en production (tests au vert et régressions peu probables).

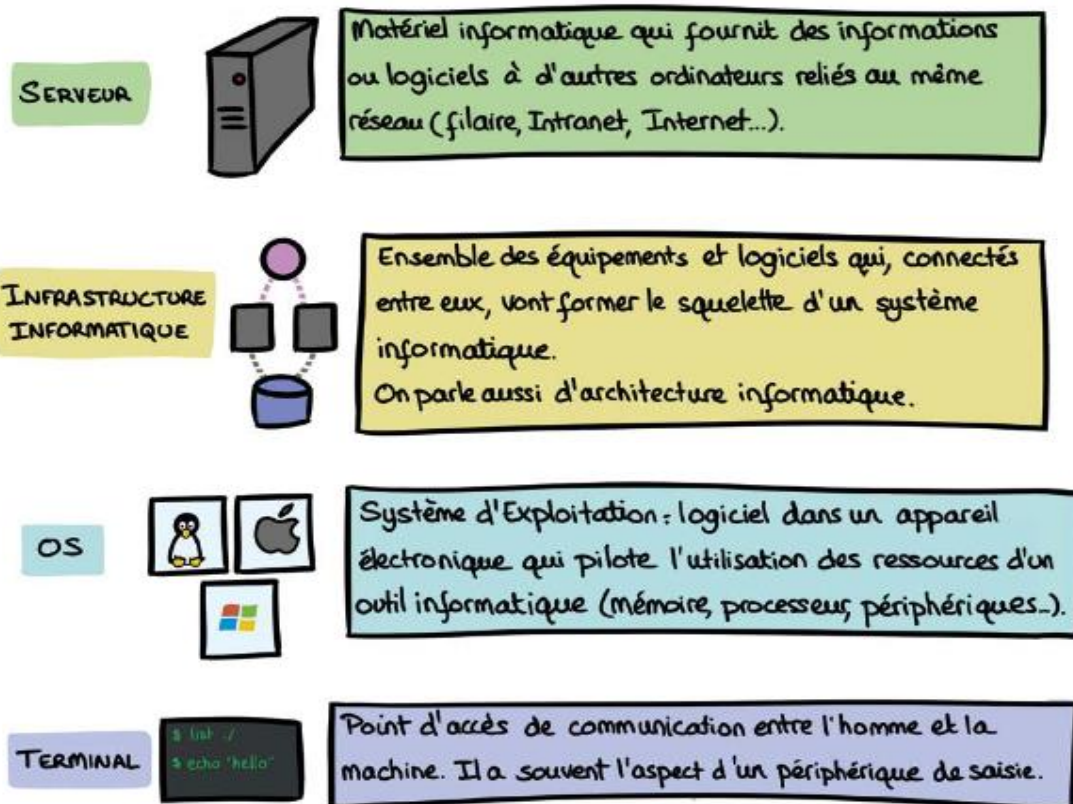
Deploy (déploiement) : phase de déploiement en production de la version codée, construite, testée, livrée.

Operate : produit en service avec le fonctionnement attendu et dans la durée, comprend l'accompagnement et le support utilisateur.

Monitor (supervision) : surveillance de la performance du système, analyse comportementale des utilisateurs lors de l'utilisation du produit, introspection sur la phase d'itération.

# Livraison continue /Devops

Cela change la façon de fonctionner dans les équipes, et nécessite de nouvelles compétences et plus de collaboration.



Source : OCTO publication

# Adaptation des infrastructures

Elles doivent s'adapter au fonctionnement de la livraison continue : les infrastructures cloud les plus populaires sont de différents types.

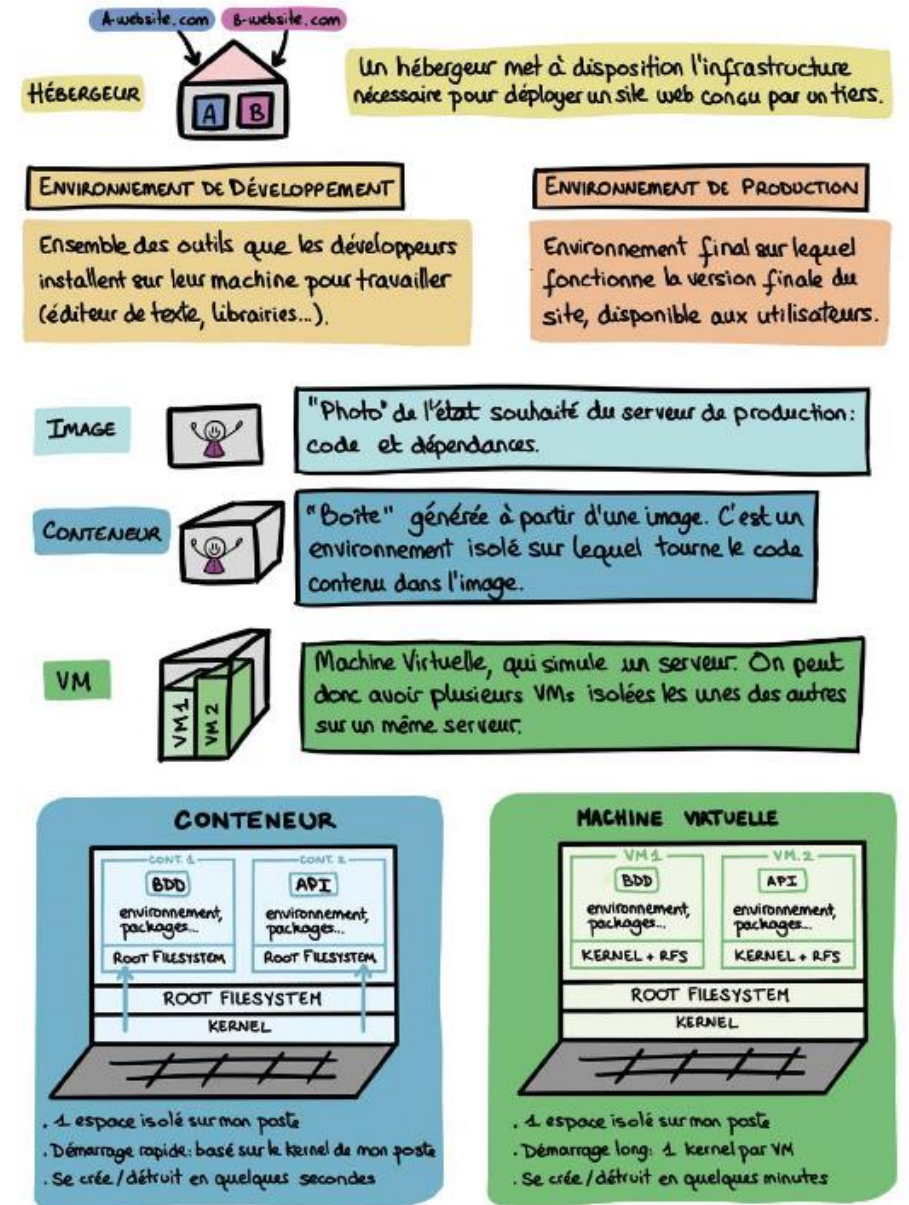
- **IaaS** (Infrastructure as a Service)
- **PaaS** (Plateforme as a Service)
- **SaaS** (Software as a Service)

## Analogie avec la pizza

- IaaS, vous créez votre pizza, vous achetez et utilisez l'ensemble des ingrédients puis vous vous débrouillez pour la cuisson et le service.
- PaaS (met à disposition) vous passez commande et vous vous faites livrer votre pizza. A vous fournir, les couverts, les boissons et les chaises.
- SaaS, (le consommateur l'utilise) : vous allez à la pizzeria et vous ne vous occupez de rien (vous payez pour l'ensemble des prestations).



- Offre IaaS** (Infrastructure as a Service), seule l'infrastructure est fournie : le consommateur est responsable de son infrastructure, de son produit et de sa configuration.
- Offre PaaS** (Plateforme as a Service) met à disposition, pour une filière technologique donnée, un environnement d'exécution dans le Cloud ainsi que des services de déploiement et d'exploitation d'applications.
- Offre SaaS** (Software as a Service) : le fournisseur est responsable de l'infrastructure, du produit et de sa configuration. Le consommateur l'utilise.



Source : OCTO publication



## Pour aller plus loin

- Offres de Services consultables sur Spote :
  - Offre Gitlab livraison continue, offre Iaas et offre Paas ...
  - Offre d'accompagnement à la livraison continue.
- Les apéros Gitlab : replay disponible ici [Apéro Gitlab](#)
- Formation de sensibilisation à la livraison continue proposée par la DNUM MSP-DAM.
  - Le 9 juin et le 12 octobre 2023 [inscription ici](#) .
- Définition d'un parcours de formation à destination des développeurs.

# Pour les experts métiers qui souhaitent découvrir la méthode en se lançant pas à pas sur un produit existant (legacy) ou un nouveau produit, PNM et MSP/DAM vous accompagne :



[Réalisation d'un produit numérique métier](#)

[Sprint 0 : aide au lancement de produit numérique](#)

[Accompagnement de la livraison continue](#)

[Gitlab](#)

[OpenStack : Infrastructure as a Service \(IaaS\)](#)

[Eco Compose : orchestrateur de conteneurs Docker](#)

[Sonarqube : forge logicielle et livraison continue](#)

[Clausier](#)

À vos questions

---

# Merci :)