



DevOps

16/10/2025

Rémi ROMMELARD



## Qui suis-je ?

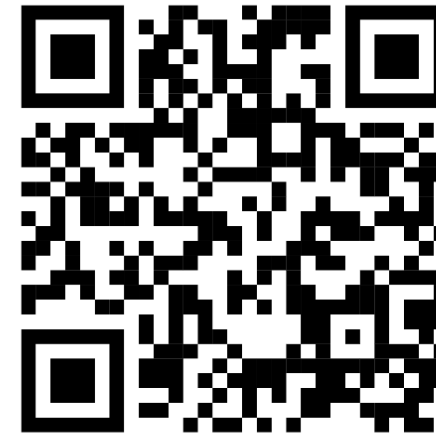


ROMMELARD Rémi  
Consultant (ACTEAM-IT)

### Formation initiale:

- BAC +3 Licence Science humaine Psychologie
- BAC +2 BTS informatique de gestion
- BAC +5 Chargé de projet en système informatique appliqué

### Parcours Professionnel:



Qui se voit côté « Dev » ? Côté « Ops » ? Les deux ?

Notre Objectif : repartir avec une vision claire, des pistes concrètes pour se lancer, et des questions répondues en direct.





DevOps, Ce n'est pas un intitulé de poste unique !

culture + pratiques + outillage qui rapprochent Dev et Ops pour livrer plus souvent, plus sûr, avec plus de valeur.

c'est une **façon de travailler** adoptée par plusieurs métiers (dev, admin/sysops, SRE, cloud, platform engineer...).

DevOps ≠ “un métier”, mais une **philosophie d'ingénierie**.



## LES 5 PILIERS : **CALMS**

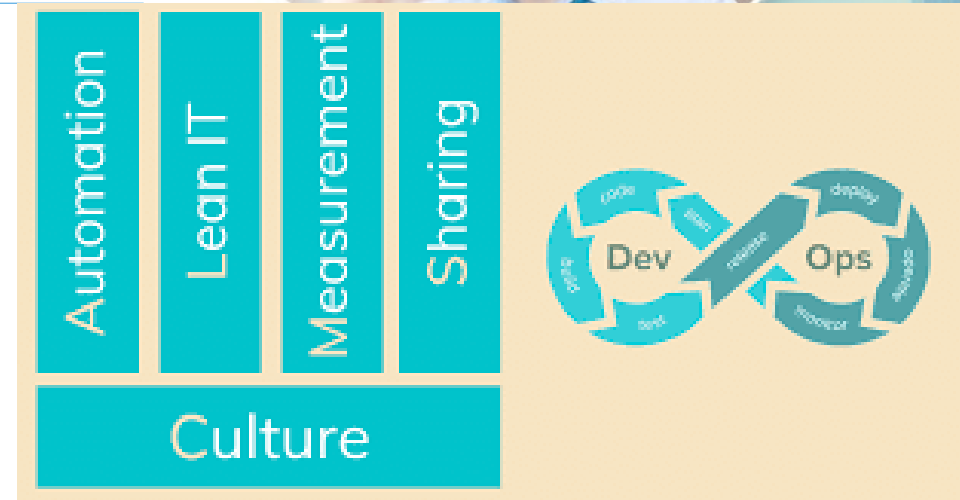
Culture : esprit d'équipe, droit à l'erreur, feedback rapide.

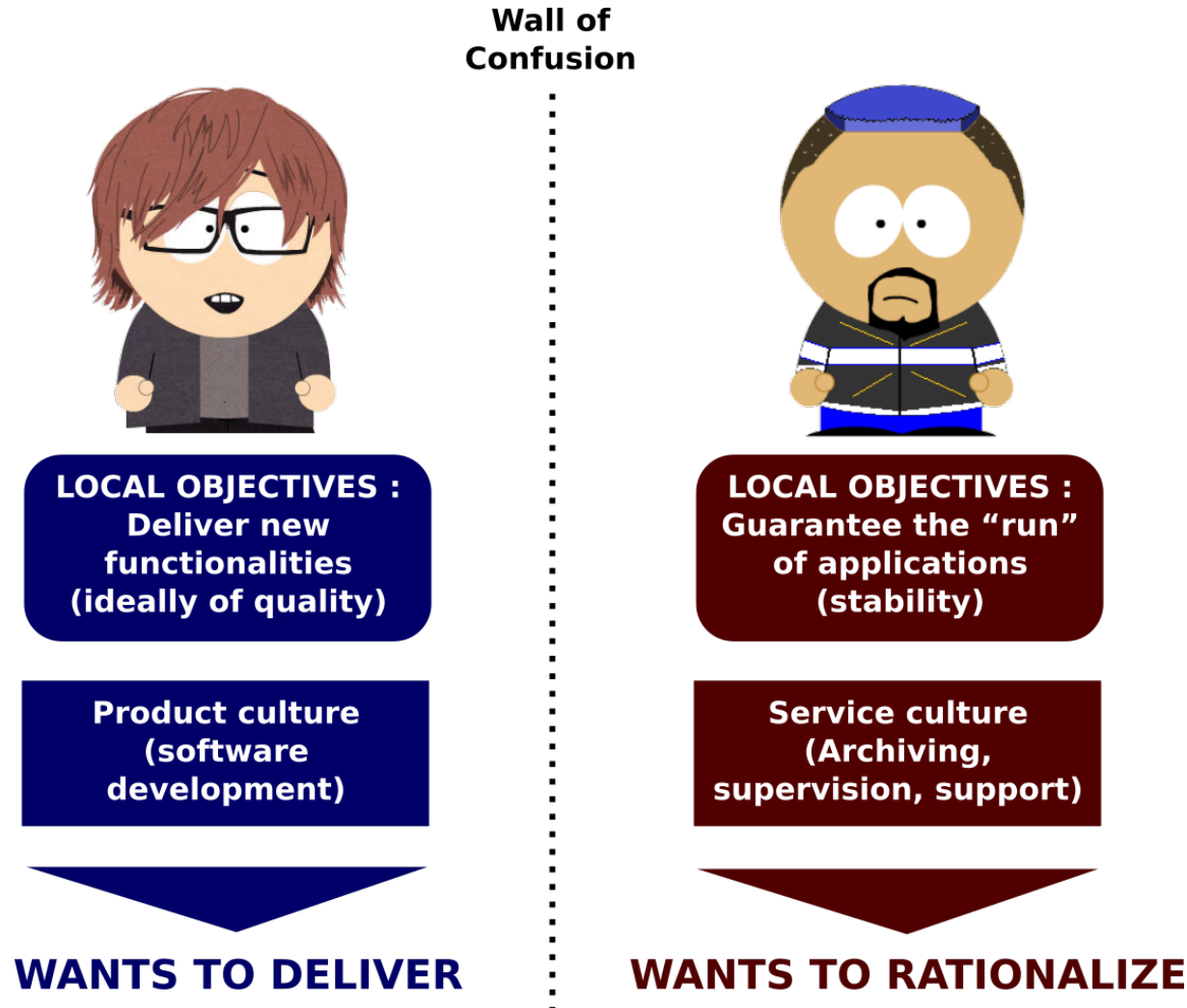
Automation : CI/CD, IaC, tests automatiques, répétabilité.

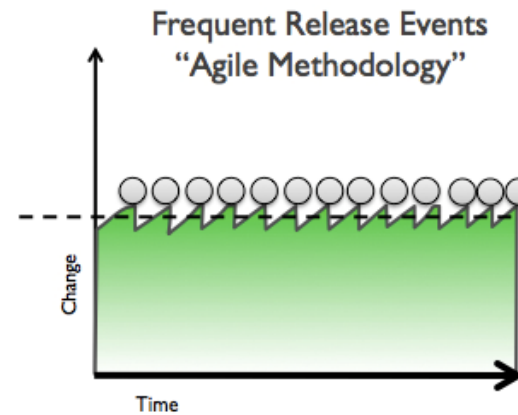
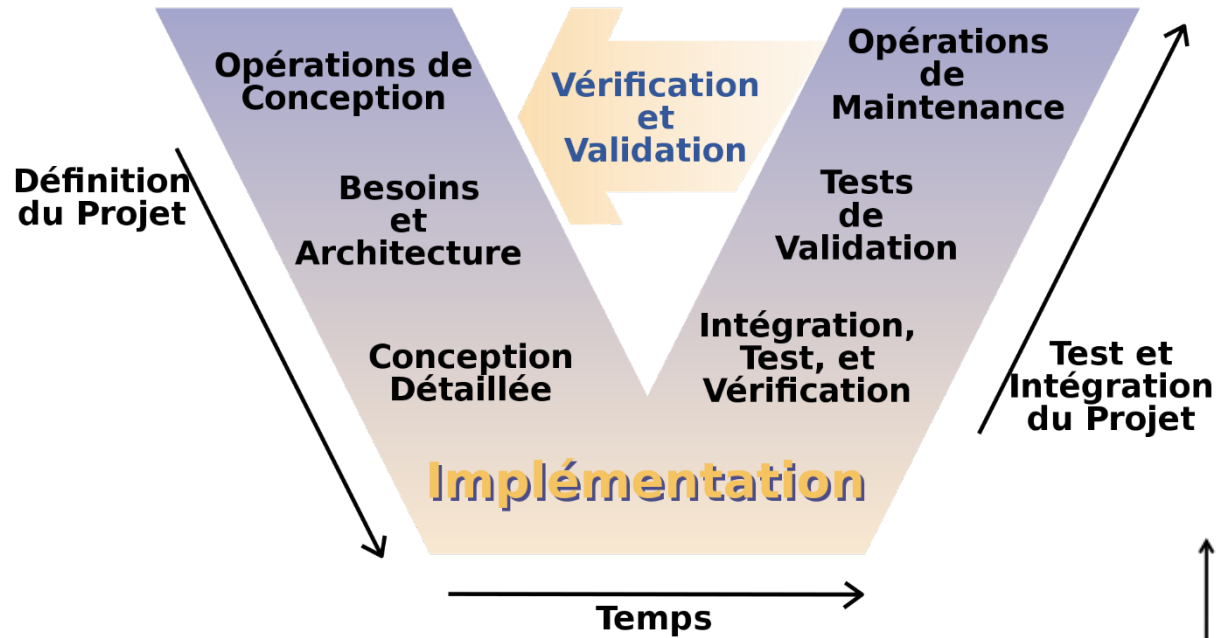
Lean : petites itérations, flux tiré, focus valeur.

Measure : métriques (lead time, fréquence de déploiement, MTTR, % échecs), SLO/SLA.

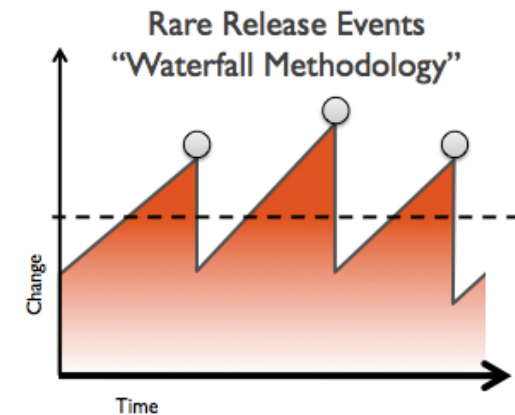
Sharing : pair-review, docs vivantes, guildes, post-mortems partagés





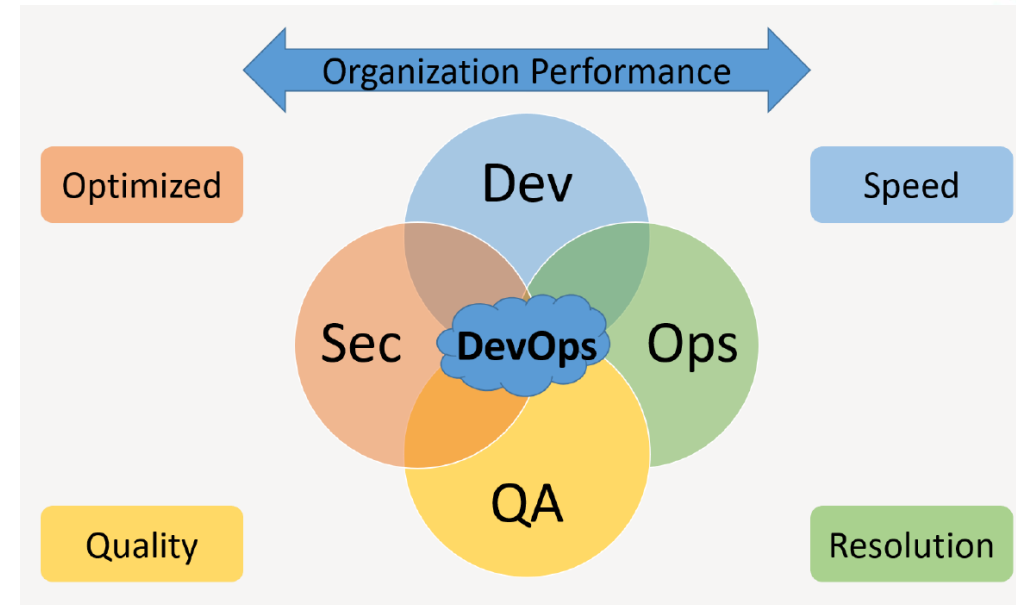
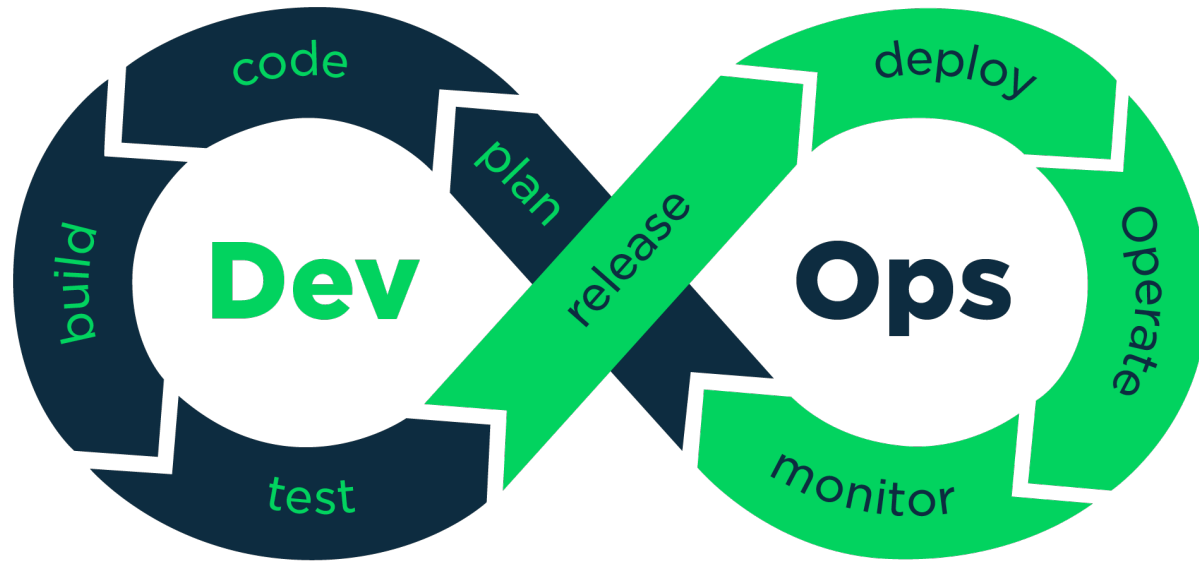


Smoother Effort  
Less Risk



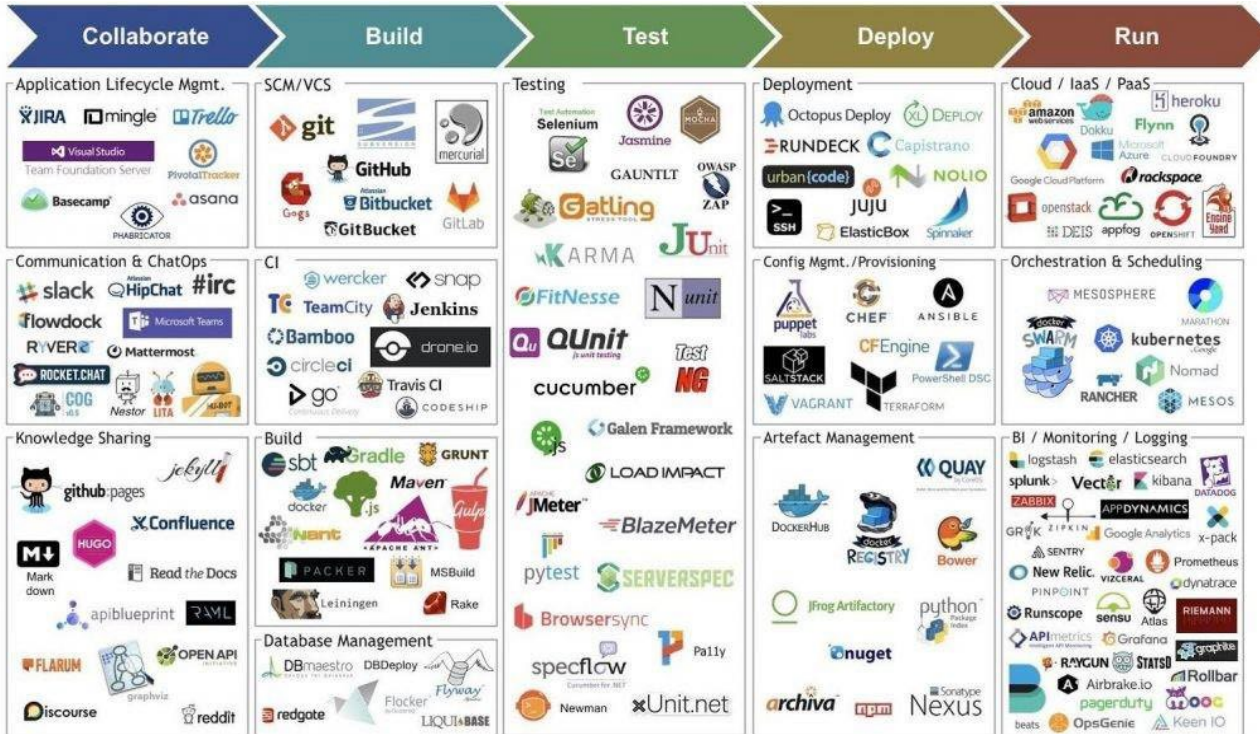
Effort Peaks  
High Risk







# Le métier



## GOAL →

- Code & VCS : Git, PR, conventions.
- CI/CD : GitHub Actions / GitLab CI / Jenkins.
- Containers & Orchestration : Docker, Kubernetes.
- IaC & CaC : Terraform, Ansible.
- Observabilité : Prometheus/Grafana, ELK/Opensearch, OpenTelemetry.
- Sécurité (DevSecOps) : SAST/DAST/Container Scan, Vault/Sealed Secrets.

## À quoi ça ressemble au quotidien ?

### Une journée type

- Revue des besoins avec une squad produit.
- Conception d'un pipeline CI/CD (build → tests → analyse sécurité → déploiement).
- Infrastructure as Code (Terraform/Ansible) + Containers (Docker/K8s).
- Observabilité : logs/metrics/traces, alerting.
- DevSecOps : scanners SAST/DAST, durcissement, secrets management.
- Doc/veille et un peu de « réunionite » 😊



# IA ↔ DevOps : dans les deux sens

GOAL →

## IA → DevOps (AIOps, Copilot, sécurité)

- Bootstrap plus vite : générer des scaffolds (CI/CD, IaC, tests)... puis PR & review humaine.
- Ops Copilot : requêtes K8s/Cloud en langage naturel, génération/édition de runbooks, requêtes logs guidées.
- AIOps : détection d'anomalies, corrélation & déduplication d'alertes, RCA assistée.
- Sécurité intégrée : revues IaC/policies, secrets/dépendances à risque, durcissement.

## DevOps → IA (MLOps / Platform)

- CI/CD/CT pour les modèles : build images, tests, model registry, canary/A-B, rollback.
- Pipelines reproductibles : données versionnées (DVC), features store, traçabilité data→model→serving.
- Infra élastique : GPU/TPU autoscaling, job queues, séparation entraînement/serving.
- Observabilité ML : drift (données/modèle), qualité, latence, coût ; alertes & playbooks.
- Gouvernance & sécurité : PII, conformité, guardrails, validation humaine avant prod.

**Règle d'or : Use AI to draft, engineers to decide.**



## Métier sans réelle formation proprement dite, ce sont les Compétences qui sont au cœur :

**Tech /produit** : Linux, réseaux, Git, CI/CD, Docker/K8s, Terraform/Ansible, scripts (Bash/Python), bases cloud (AWS/GCP/Azure/OVH).

**Comportemental** : collaboration, curiosité, autonomie, partage, remise en question.

Il s'agit plus de profils différents qui se rejoignent sur cette philosophie (augmenter la qualité de l'usine pour produire un outil de qualité),

**Vous allez avoir besoin de comprendre tous les silos et le produit, donc il faut être ouvert en plus d'aimer à passer du temps sur son PC ;)**





# Vision du métier étudiant/vie professionnelle

- L'apprentissage du « métier » passe avant tout par une veille importante
  - Faites de l'Infra as Code (Terraform / Ansible / Docker / Kubernetes)
  - Tester des outils
  - Lisez des REX (abonnez-vous à des flux RSS)
  - Passez des certifications (Linux fondation par exemple)
  - Soyez le plus propre possible dans votre travail (pourquoi je le fais, comment je le fais)
- L'école vous enseignera les bases théoriques, approfondissez et spécialisez-vous ! (automaticien, cloud, testing, sécurité...)
- Vos contraintes peuvent être budgétaire et technique
  - Il faut s'adapter à son contexte et ne pas forcer les choses
  - Il faut être pro-actif, identifier les failles et être force de proposition
  - L'Open Source est un terrain de jeu incroyable (mais ne veut pas dire gratuit ;) )
  - Les produits portés par la Linux Fondation et la Cloud Native Computing Fondation (entre autres) sont des valeurs sûres sur le marché.



## Qu'apporte le métier?

---

- Des missions et des périmètres variés
- Une plus grande confiance en soi
- Au cœur du système d'information
- Pour l'entreprise, une plus grande stabilité de son SI et respects des bonnes pratiques



- Croire que « installer Kubernetes = DevOps ».
- Faire une factory d'outils sans usage produit.
- Oublier les métriques (on ne peut pas améliorer ce qu'on ne mesure pas).



# Quelles sont les perspectives d'évolution du métier

**DevOps = méthode, pas un intitulé**

## Rôles praticiens (IC)

- SRE : SLO/SLA, fiabilité, réduction du toil, MTTR.
- Platform Engineer : plateformes internes, GitOps, devX.
- Cloud/Infra Architect : design, sécurité, coûts (FinOps).
- DevSecOps Engineer : sécurité intégrée, politiques, scanning.
- Observabilité/Perf : logs/metrics/traces, APM.
- Data/ML Platform (MLOps) : registry, déploiements modèles, drift.

### Mini-KPIs qui prouvent l'évolution

- Lead time ↓, fréquence déploiements ↑,
- MTTR ↓, taux d'échec ↓,
- € infra / feature ↓

## Leadership / produit

- Tech Lead / Chapter Lead : standards, mentoring.
- Engineering Manager / Platform/SRE Manager : people + delivery.
- TPM / Delivery Manager : flux, dépendances, risques.
- Head of Platform/SRE → CTO (long terme) : stratégie, budget, gouvernance.

### Roadmap réaliste

- 0–1 an : Linux, Git, CI/CD, Docker, IaC ; pipeline bout-en-bout + observabilité basique.
- 1–3 ans : ownership service/plateforme, SLO+alerting, GitOps, sécurité intégrée ; mentoring.
- 3–5 ans : piloter une plateforme, standards & budgets (FinOps), gouvernance, lead/manager.

**Évoluer = augmenter l'impact, pas collectionner des outils**

- Scope : d'un pipeline/équipe → à une plateforme/organisation.
- Autonomie : du runbook appliqué → à la définition des standards.
- Influence : de la PR locale → à la vision et la gouvernance.



A group of four people are gathered around a table in what appears to be a meeting or collaborative work environment. On the left, a man in a light blue shirt is looking down at a tablet. Next to him, a woman with long dark hair is also looking at a device. In the foreground, a man in a bright blue polo shirt is pointing at a smartphone held by a woman with blonde hair. The background is slightly blurred, showing more of the office setting. A semi-transparent white box is overlaid on the left side of the image, containing the text 'Questions/Réponses'.

## *Questions/Réponses*