

Olivier
Philippot

DevGreenOps : Opportunité de repenser la culture du test pour plus d'agilité et de collaboration

11 JUIN 2024
BEFFROI DE MONTROUGE



JOURNÉE
FRANÇAISE
DES TESTS
LOGICIELS

Green et DevOps : Contexte

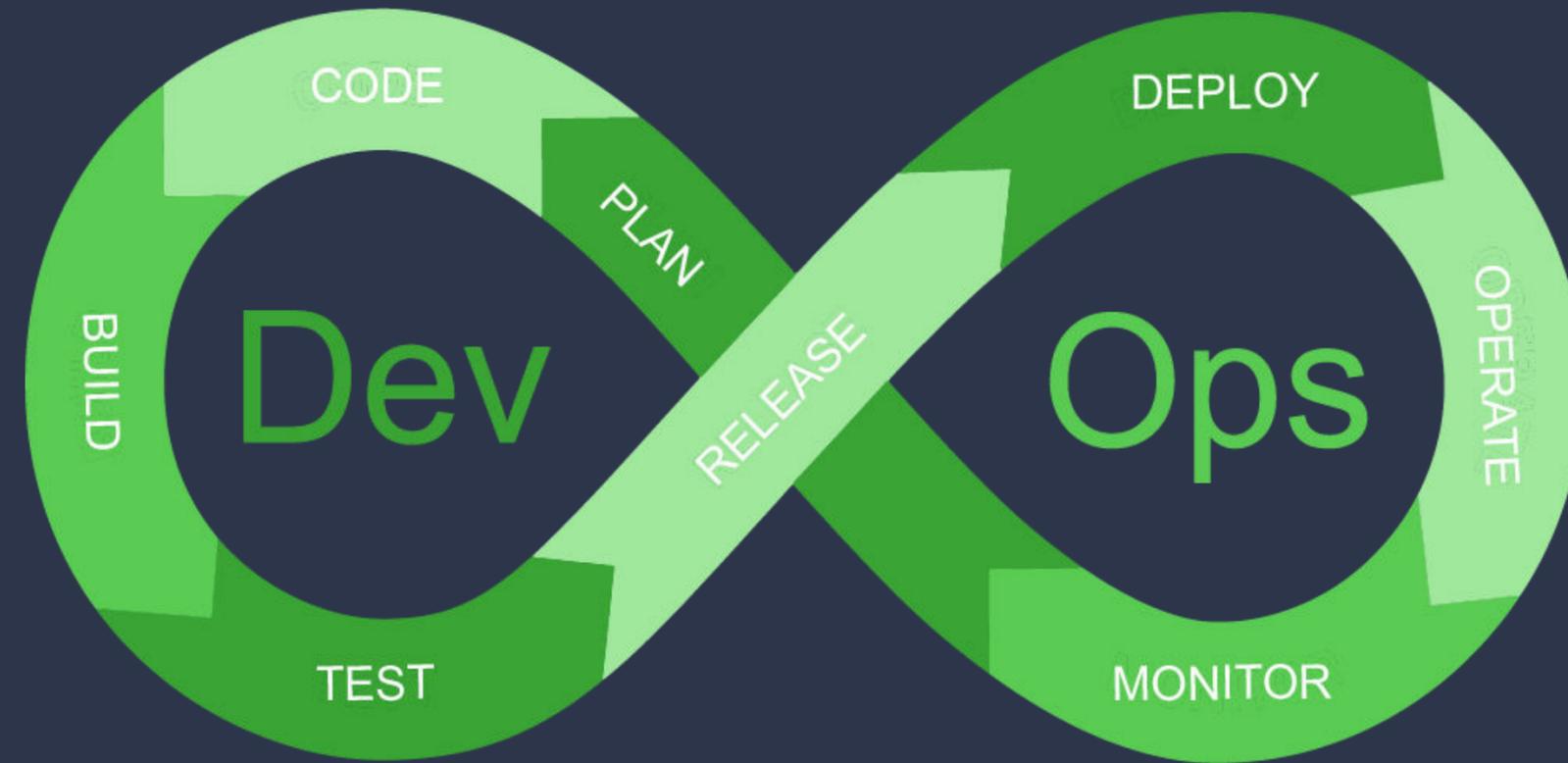
Définition du DevGreenOps

Opportunités du DevGreenOps pour le test

Nouvel outillage de test et de mesure

Conclusion : Les challenges du DevGreenOps pour les équipes de test

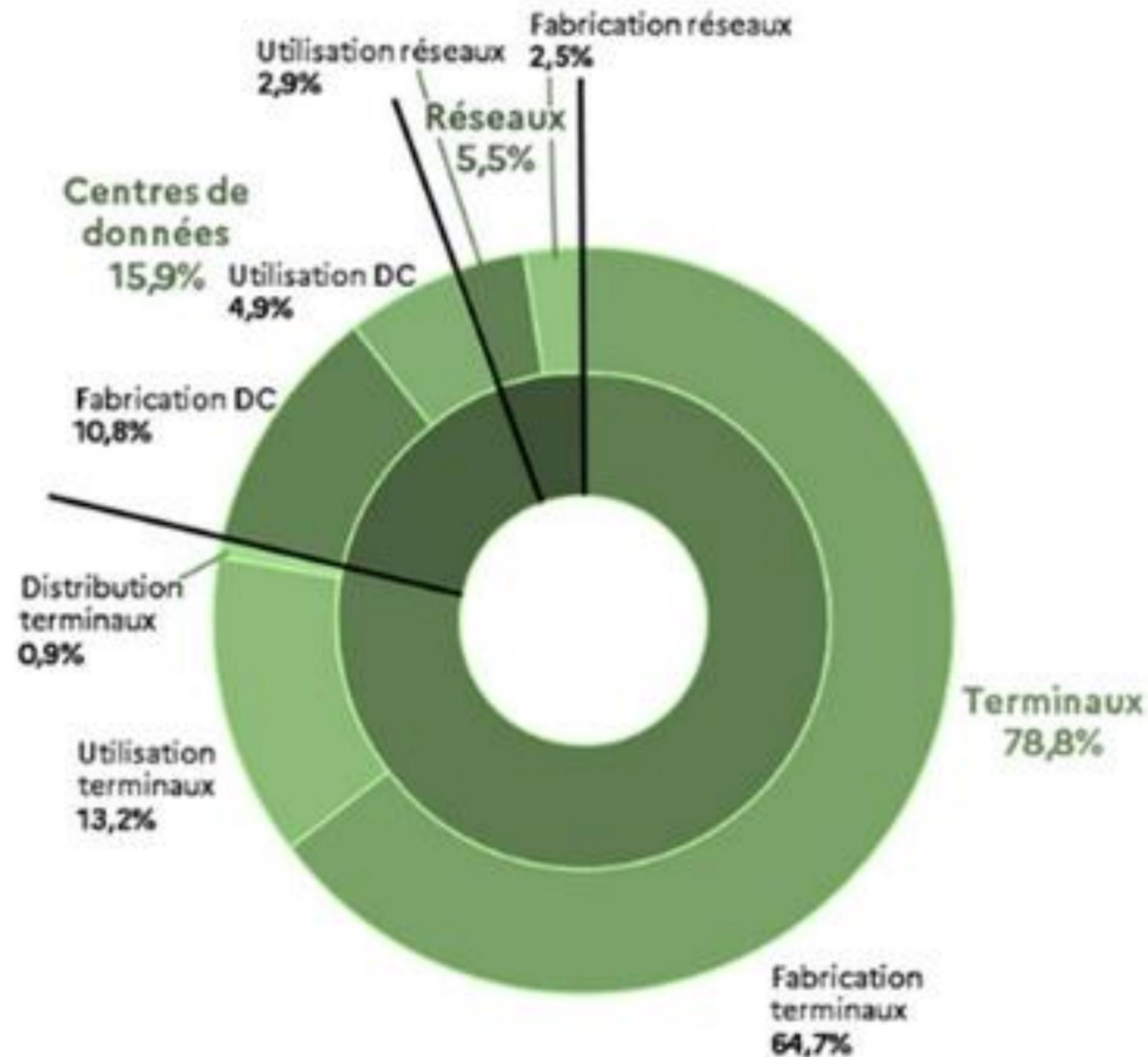
Contexte du DevGreenOps



- **Numérisation croissante** des services ➡ Besoin de **livrer plus** de produits logiciels, avec plus de fonctionnalités, plus fréquemment...
- Méthode Agiles + IC/DC ➡ Avènement de la culture DevOps
- **Accélération des livraisons pour une adaptation aux demandes clientes**

2,5%
des émissions
nationales en 2020
soit 16,9 MtCO₂eq.

Une croissance en
tendanciel de
+45% en 2030
par rapport à 2020



Etude ADEME-ARCEP (2022-2023)

Matériel plus
puissant

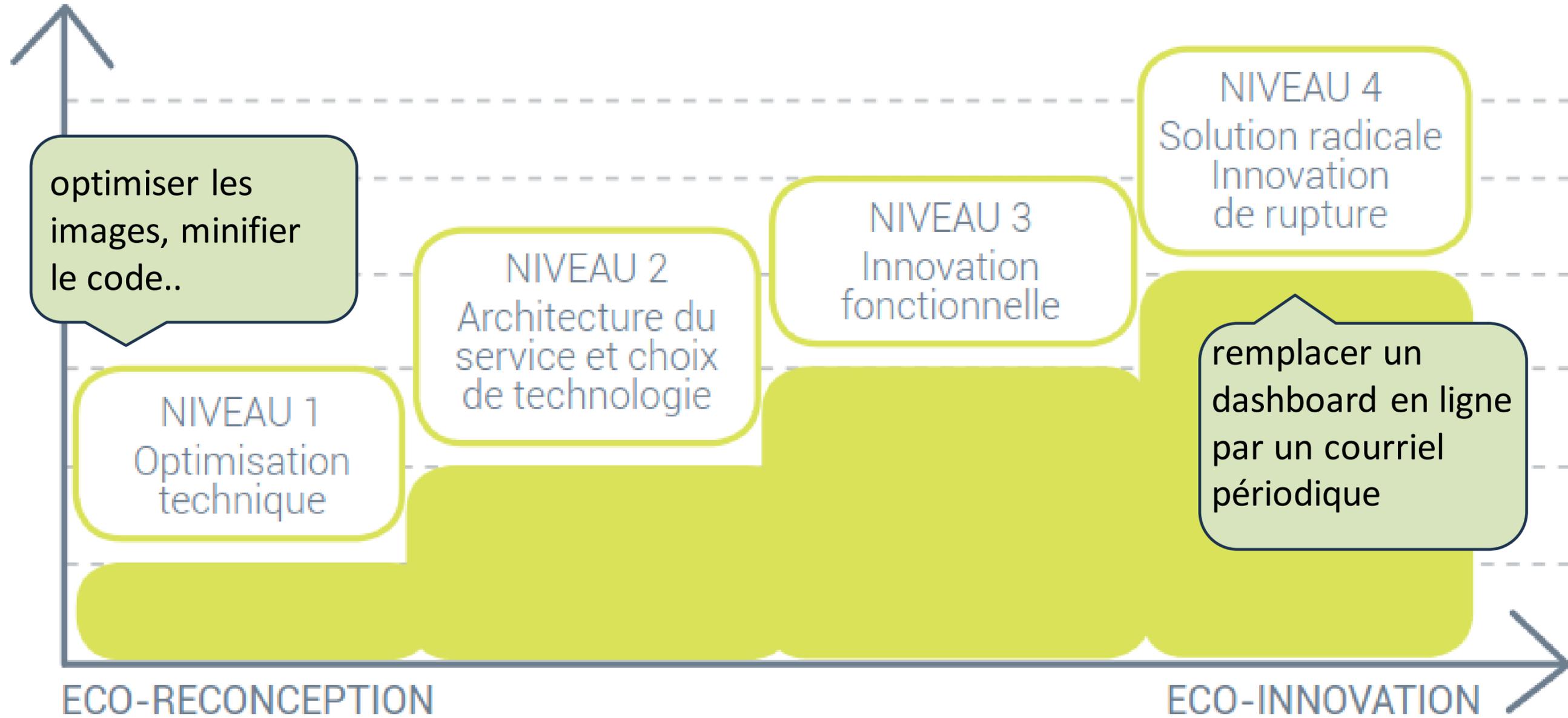
Logiciel plus
lourd

Ralentissement

Remplacement
matériel

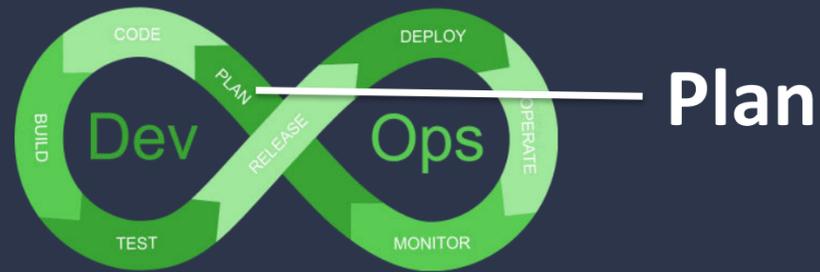
**Le "sac à dos" écologique : 16
kg de matières premières pour
fabriquer 1 g de circuit
électronique.**

ECO-EFFICACITE



Le DevGreenOps

Le **DevGreenOps** est une culture permettant d'intégrer l'aspect environnemental dans le développement des services numériques via des pratiques de mesure et d'amélioration continue de l'impact de l'application.



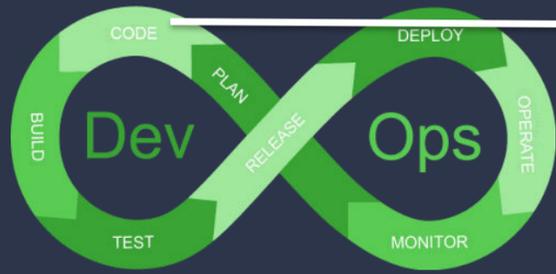
- Phase critique : rétrospective des itérations précédentes, choix des éléments à tester...
- Planification de pratique de tests agiles

Pratiques de tests adaptées au DevGreenOps

Développement piloté par les tests d'acceptation (ATDD) : Proche de l'utilisateur et applicable du staging à la production, donc permettra de détecter des anomalies énergétiques côté utilisateur

Tests exploratoires : Permet de confirmer des intuitions ou des risques, d'évaluer des pratiques de dev, de modifier des paramètres influençant la consommation (luminosité, réseau GSM...)

Code



- Test de choix technique (Choix de benchmark...)
- Analyse de code sur des pratiques Green

Plugin SonarQube Ecocode

The screenshot shows the SonarQube interface for a project named 'kube-demo'. The main content area displays a list of code quality issues for the file 'src/.../com/cacib/unicom/demo/service/RefDataServiceImpl.java'. The issues are categorized as 'Code Smell' and are mostly 'Minor' in severity. The interface includes a left-hand navigation menu with filters for Type, Severity, Scope, Resolution, Status, Security Category, Creation Date, Language, Rule, Tag, and Directory. The top right corner shows the date 'September 30, 2022, 9:30 AM' and 'Version 3'. The bottom right corner shows '1 / 8 Issues' and '1h effort'.

Issue Description	Severity	Effort	Tags
Remove this useless assignment to local variable "result". Why is this an issue?	Major	15min effort	cert, cwe, unused
Remove this unused "result" local variable. Why is this an issue?	Minor	5min effort	unused
The variable result is not assigned. Why is this an issue?	Minor	5min effort	eco-conception
The variable k is not assigned. Why is this an issue?	Minor	5min effort	eco-conception
Remove this useless assignment to local variable "result". Why is this an issue?	Major	15min effort	cert, cwe, unused
Remove this unused "result" local variable. Why is this an issue?	Minor	5min effort	unused
The variable result is not assigned. Why is this an issue?	Minor	5min effort	eco-conception
The variable k is not assigned. Why is this an issue?	Minor	5min effort	eco-conception

Shift right de la qualité ➡ Nécessité de passer sur des bonnes pratiques avec des gains réels

Exemple de règles "concrètes"

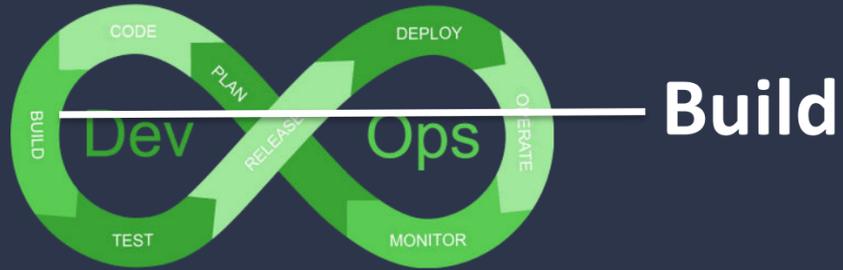
The screenshot shows three lint warnings from Android Studio. The first warning is for the file `src/main/AndroidManifest.xml` and is titled "Battery optimization should not be ignored. See Rule". It is categorized as a "Code Smell" with a "Major" severity, is "Open", and has a "1h effort" to resolve. The second and third warnings are for the file `src/.../univpau/uppamaps/Screens/City/CityListener.java` and are titled "Use com.google.android.gms.location instead of android.location to maximize battery life. See Rule". Both are categorized as "Code Smell" with a "Major" severity, are "Open", and have a "20min effort" to resolve. Each warning includes a "Comment" link and a list of tags: "ecocode, environment, power" for the first, and "ecocode, environment, optimized-api" for the others. The warnings are dated "21 hours ago".

src/main/AndroidManifest.xml

- Battery optimization should not be ignored. See Rule 21 hours ago ▾ L8 🔗 ⌵
👤 Code Smell ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 1h effort Comment 🌿 ecocode, environment, power ▾

src/.../univpau/uppamaps/Screens/City/CityListener.java

- Use com.google.android.gms.location instead of android.location to maximize battery life. See Rule 21 hours ago ▾ L5 🔗 ⌵
👤 Code Smell ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 20min effort Comment 🌿 ecocode, environment, optimized-api ▾
- Use com.google.android.gms.location instead of android.location to maximize battery life. See Rule 21 hours ago ▾ L6 🔗 ⌵
👤 Code Smell ▾ 🚨 Major ▾ 🔵 Open ▾ Not assigned ▾ 20min effort Comment 🌿 ecocode, environment, optimized-api ▾



- Lancement de test sur Staging permettant de détecter **au plus tôt** des surconsommations
- Attention à la **charge de test** ➔ Tests sur un périmètre critique et limitée

Plugin Greenspector Gitlab

Ecoscore obtenu simplement avec un test simple de benchmark (Résilience par rapport à des modification du système sous test)

Informations sur le projet

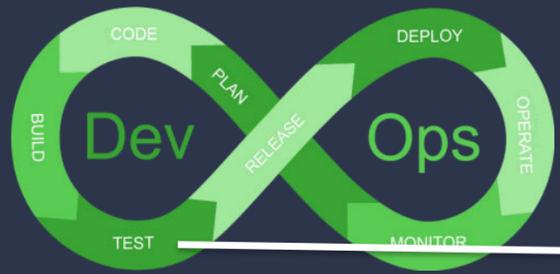


ecoscore 62

37 validations

1 branche

0 étiquette



Test

- Des nouvelles exigences non fonctionnelles à vérifier
- Des tests spécifiques (proches des tests existants mais avec des spécificités)



Refonte du site



Average Ecoscore per page

65

Average carbon impact per page

1 g eqCO₂

Average volume of data exchanged per page

1 Mo

Average number of HTTP requests per page

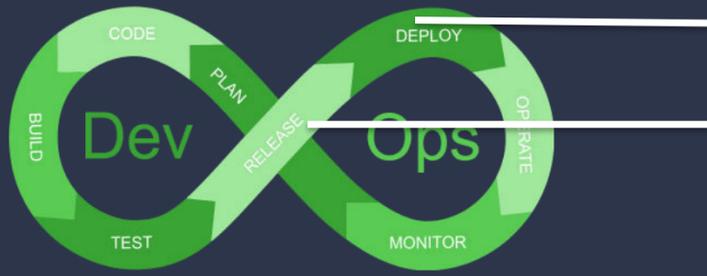
30 requêtes

Maximum energy overconsumption on Idle Foreground

20 %

Maximum energy overconsumption on Idle Background

5 %



Release & Deploy

- Capacité à détecter, via le test, des KPI menant à un no go de la release
- Tester aux extrêmes (Connexion, plateforme ancienne génération...)

Parcours fonctionnel d'une application bancaire

La connexion réseau impacte faiblement la performance jusqu'à la 3G.

En moyenne :

Wifi 4,0s (2,8s précédemment)

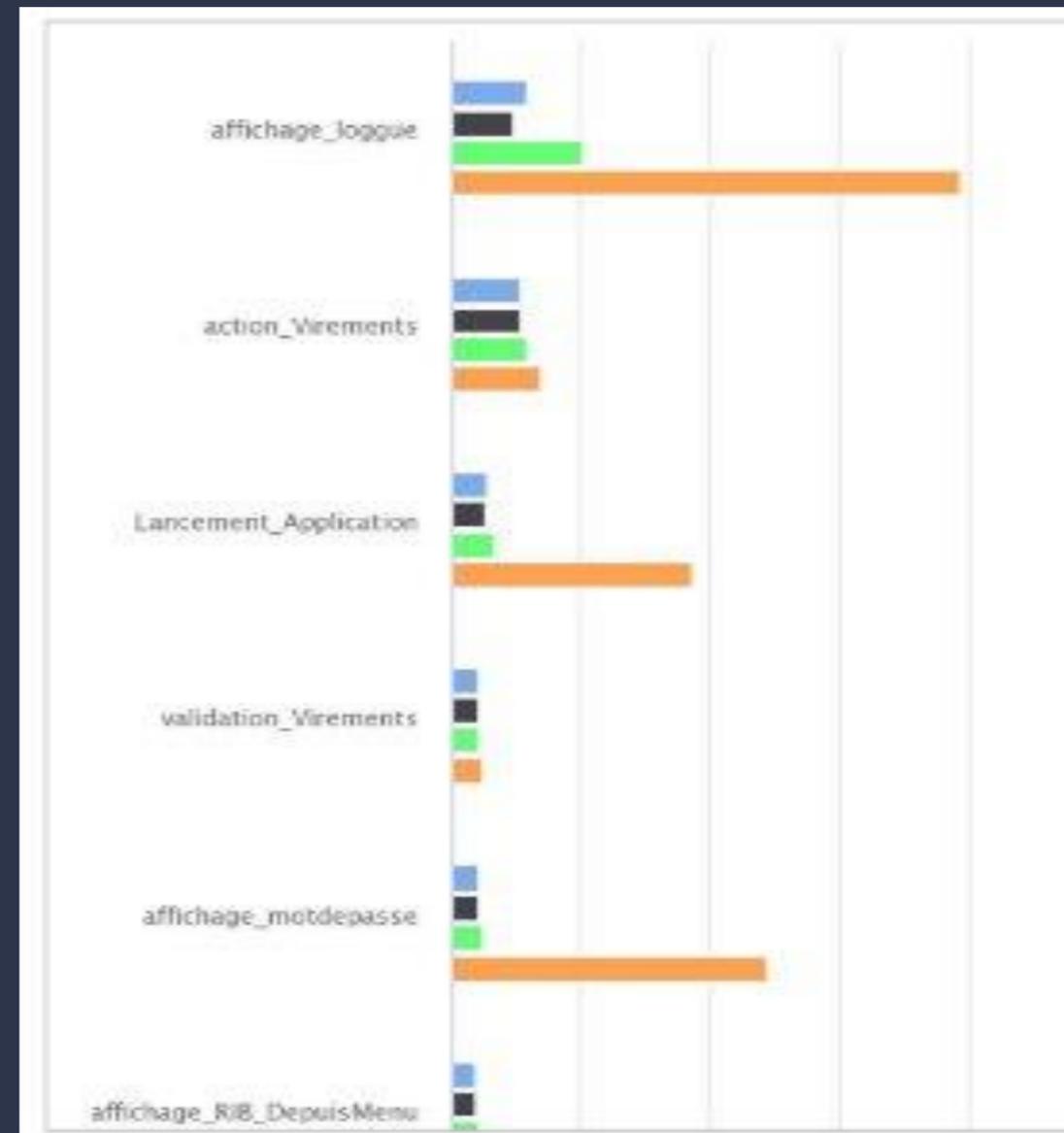
4G 3,8s

3G 4,9s (3,4s précédemment)

2G 14,6s (5,2s précédemment)

Certaines fonctions sont fortement impactées. Différence entre WiFi et 2G :

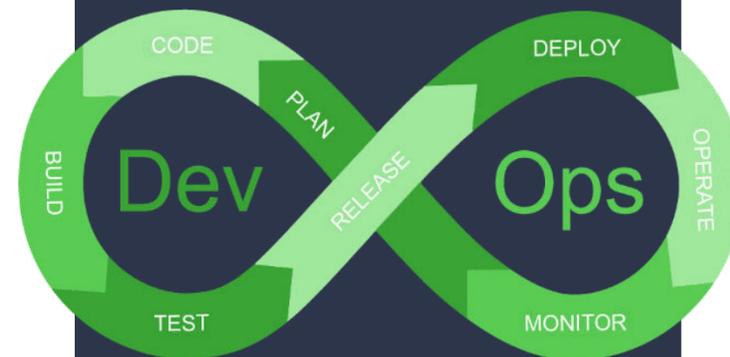
- Lancement de l'application : + 30s
- Affichage mdp : + 30s
- Affichage loggué : + 66s



Shift Left

Détection Hotspot au plus tôt

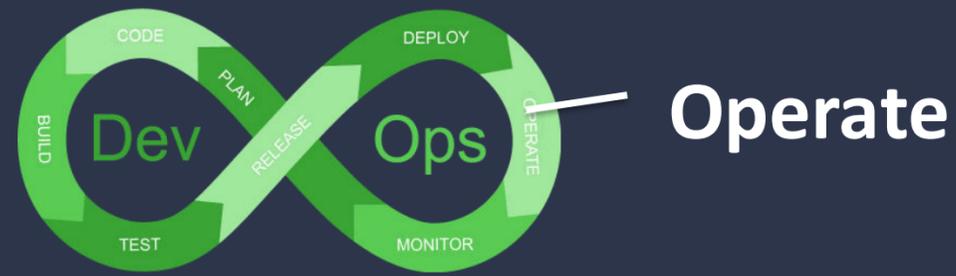
Instabilité des tests
Charge de test
Microbenchmarking



Shift Right

Evite le "it's Ok on my computer"
Proche de l'environnement cible
Analyse sur périmètre plus global
(Service tiers...)

Jeux de données
Maintenance de tests en plus
Analyse sur périmètre plus global



- Suivre les nouvelles métriques lors des tests de charges
- Mieux adapter la capacité des serveurs

Exemple de monitoring avec Scaphandre



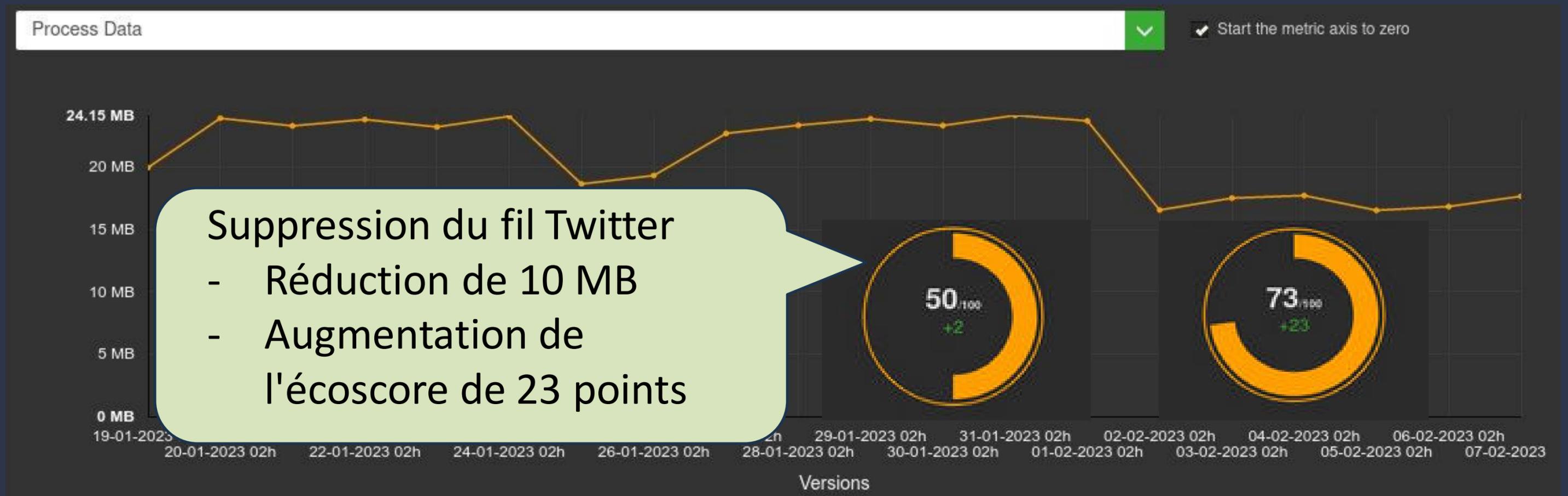


Monitor

Suivi des impacts le plus réaliste, cependant nécessite une prise en compte rapide des améliorations

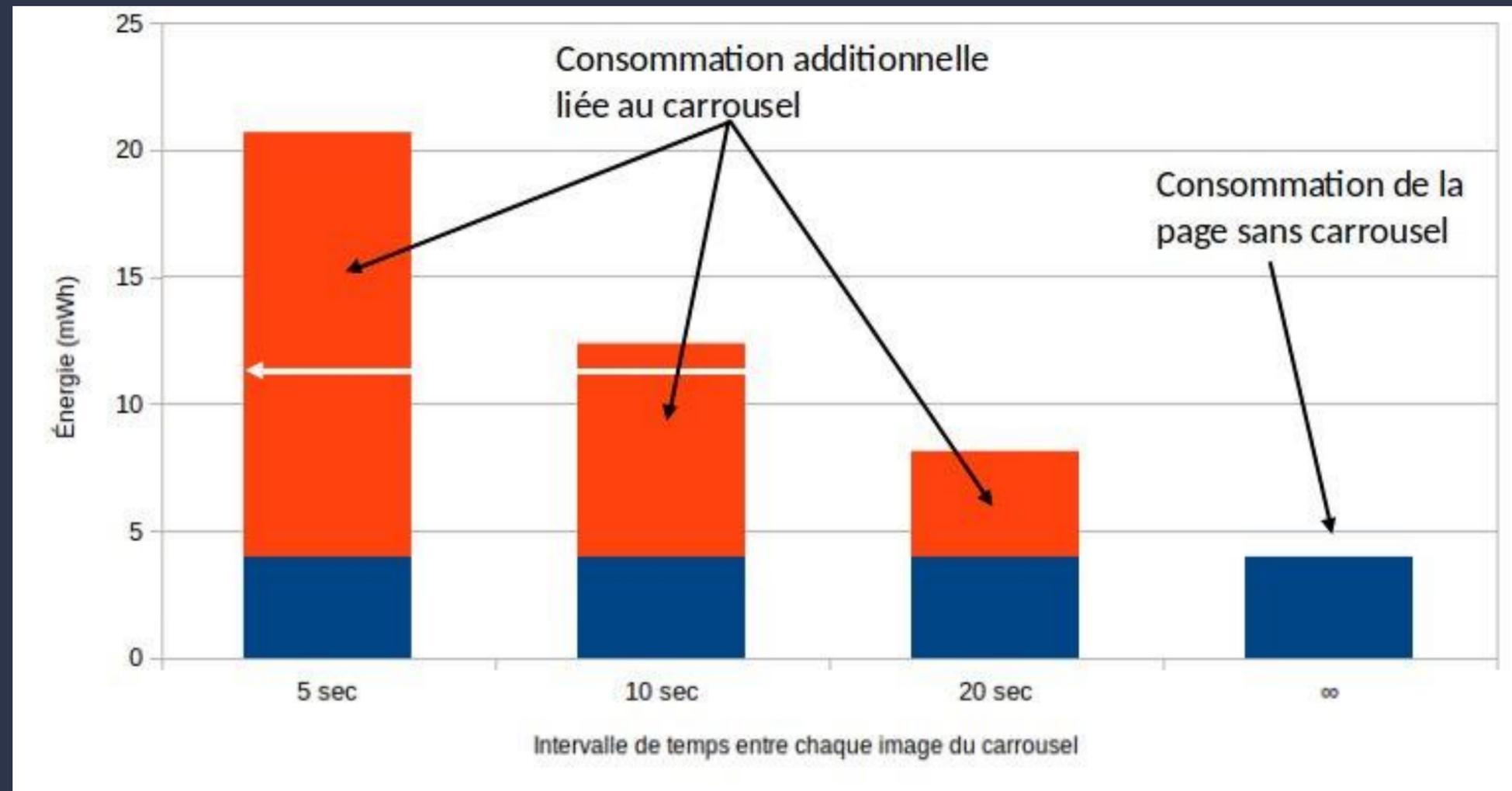


Suivi du site orange.com en production



Déploiement plus rapide + feature flipping ➡ Benchmark des pratiques simplifié

Evaluation de carrousel



Nécessité de repenser les KPI associés aux tests (Par exemple métriques de performances classiques à compléter)

Exemples de métriques

Energie, pour le terminal utilisateur et le serveur

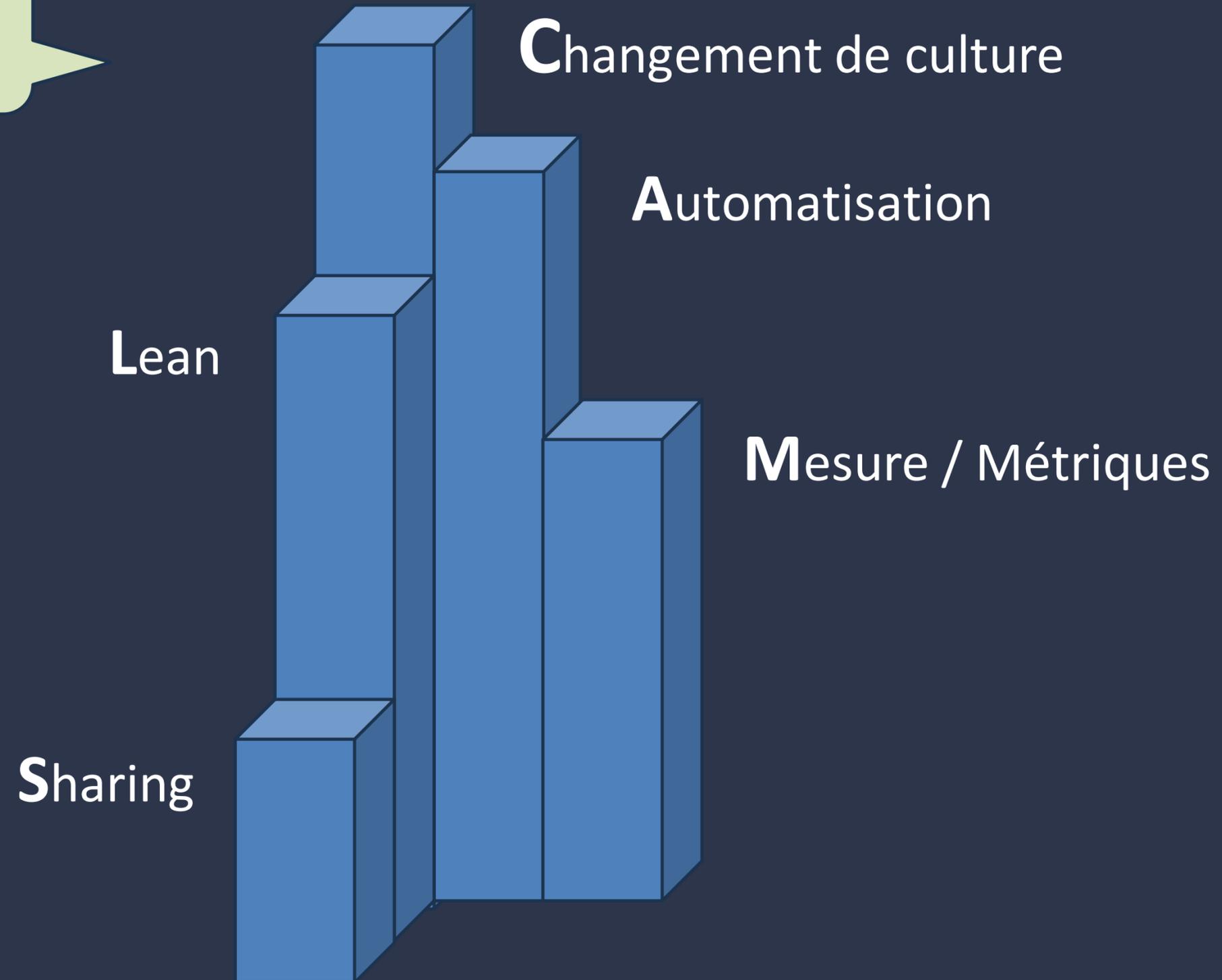
Gaz à effet de serre en CO₂e, par tiers (terminal / réseau / datacenter) et/ou avec répartition par phase de cycle de vie

Des impacts sur des critères autre que les gaz à effet de serre

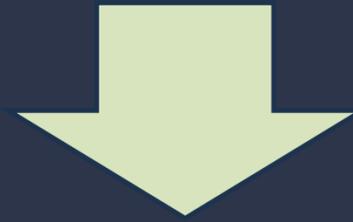
Tout score d'écoconception intégrant une prise en compte de métrique primaire (énergie, CPU, ...)

Savoir mesurer son progrès

CALMS



Des tests plus en amont, plus réguliers, sur différents environnements...
Besoin de benchmarker des pratiques, un framework...



Nécessité d'avoir un niveau d'automatisation plus élevé

CEPENDANT...

Solutions peu automatisables (Pas d'id et pas accessible)

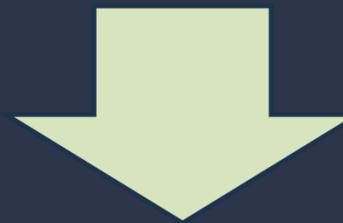
Périmètre fonctionnel qui bouge

Sécurité différente entre staging, preprod et prod

Pas de compte de test pour la prod

L'écoconception est un argument en plus pour améliorer l'automatisation dans le cadre de la culture DevOps

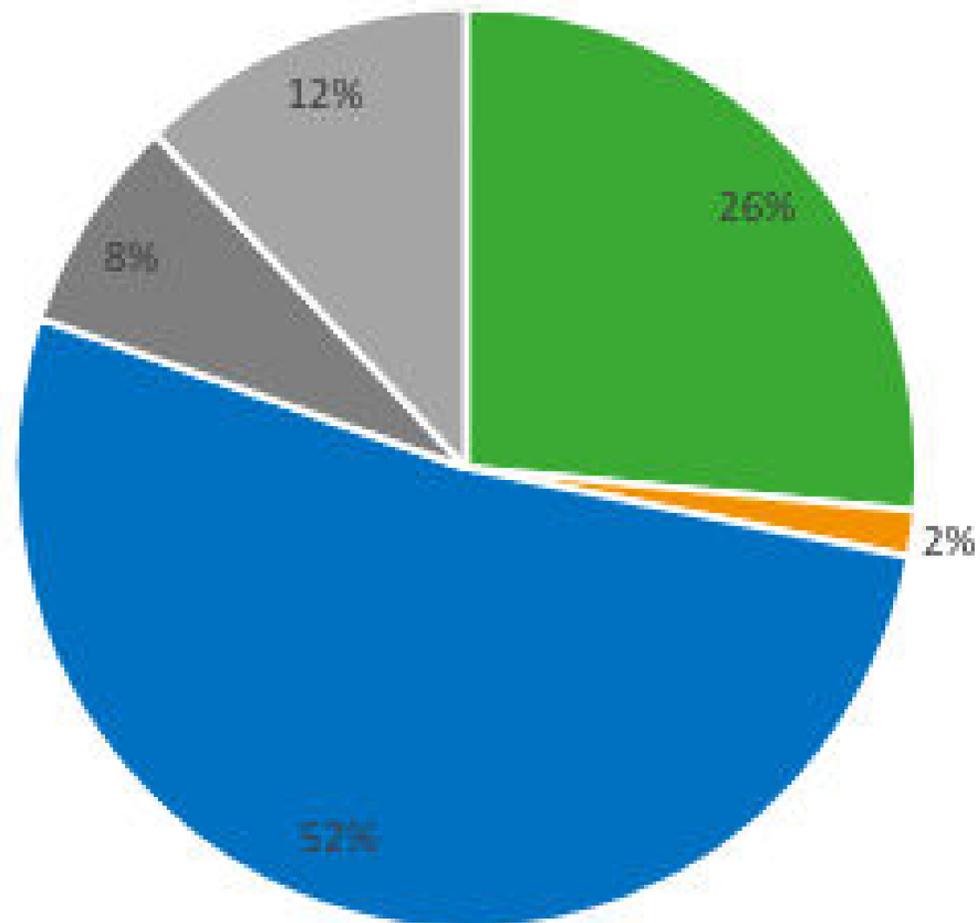
- Passage d'objectif de couverture à des objectifs de détection de l'impact réel
- Faire de la place pour les tests Green (mais aussi accessibilité...)
/ Ne pas prendre plus de temps pour l'organisation



- Trouver ce que l'utilisateur fait réellement (Usage d'analytics...)
- Penser défaillance (abandon de panier, rechargement...)
- Penser gaspillage : fonctions non utilisées par l'utilisateur > maintenance non inutile de test et de code > Réduction des efforts de l'organisation

Opportunités
de prioriser
les tests

Priorisation des tests par rapport à
l'impact réel



- Smartphone - Making and usage - Mix 7% France - 93% World
- Tablet - Making and usage - Mix 7% France - 93% World
- PC - Making and usage - Mix 7% France - 93% World
- Network - Making and usage - Mix 0% France - 100% World
- Server - Making and usage - Mix 0% France - 100% World

Les équipes ne sont pas toujours prêtes et matures par rapport à une démarche totalement Devops ➡ Nécessité de "Lean & Sharing"

AIRFRANCE 

Améliorer l'interface avec les équipes de dev

Le Service Center Test propose une offre packagée pour les projets : un service outillé comprenant la conception, l'exécution et l'analyse des tests, avec un accompagnement dans la durée pour le suivi de l'amélioration des projets.

Les projets adhèrent à la démarche :

« Effectivement, on l'a implémentée un peu brute cette modification. On va rectifier dans la prochaine release. »

Tech Lead, au vu des surconsommations détectées par les Testeurs

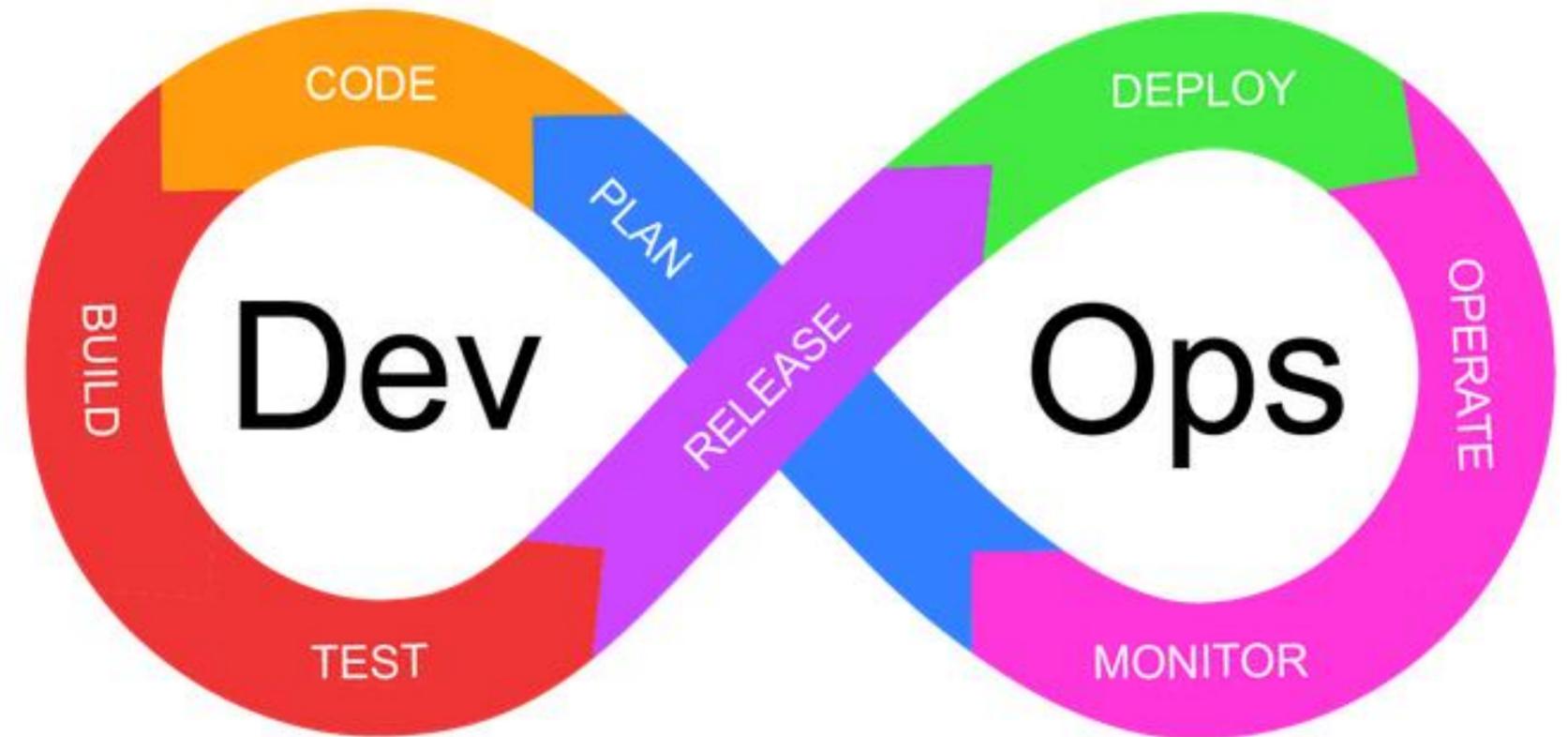
- Nécessité de suivre et prouver que les efforts portent leurs fruits
- ➔ **Mise en place de métriques spécifiques**
- Pilotées dans un Sucess Plan, Copils, rétrospectives...
- Ce ne sont pas des métriques techniques (Taux d'équipes ayant mis un OKR Green, Taux de correction...)

Exemples de mise en place de pratique de Gamification dans une banque

- Mise en place d'un label interne
- Les squads ne sont pas forcées à mettre en place des tests d'acceptance Green, par contre la gamification du label permet d'inciter à la participation
- Engagement progressif des squads
- Atteinte d'une limite d'optimisation par certaines squads qui amènent à aller challenger les hotspots dans les services tiers

Les challenges du DevGreenOps

- **Mettre en place un outillage adéquat**
- Certains outils sont compatibles avec cette approche.
- Attention à ne pas verdir un outil de perf existant.



- Travailler sur les impacts réels des solutions

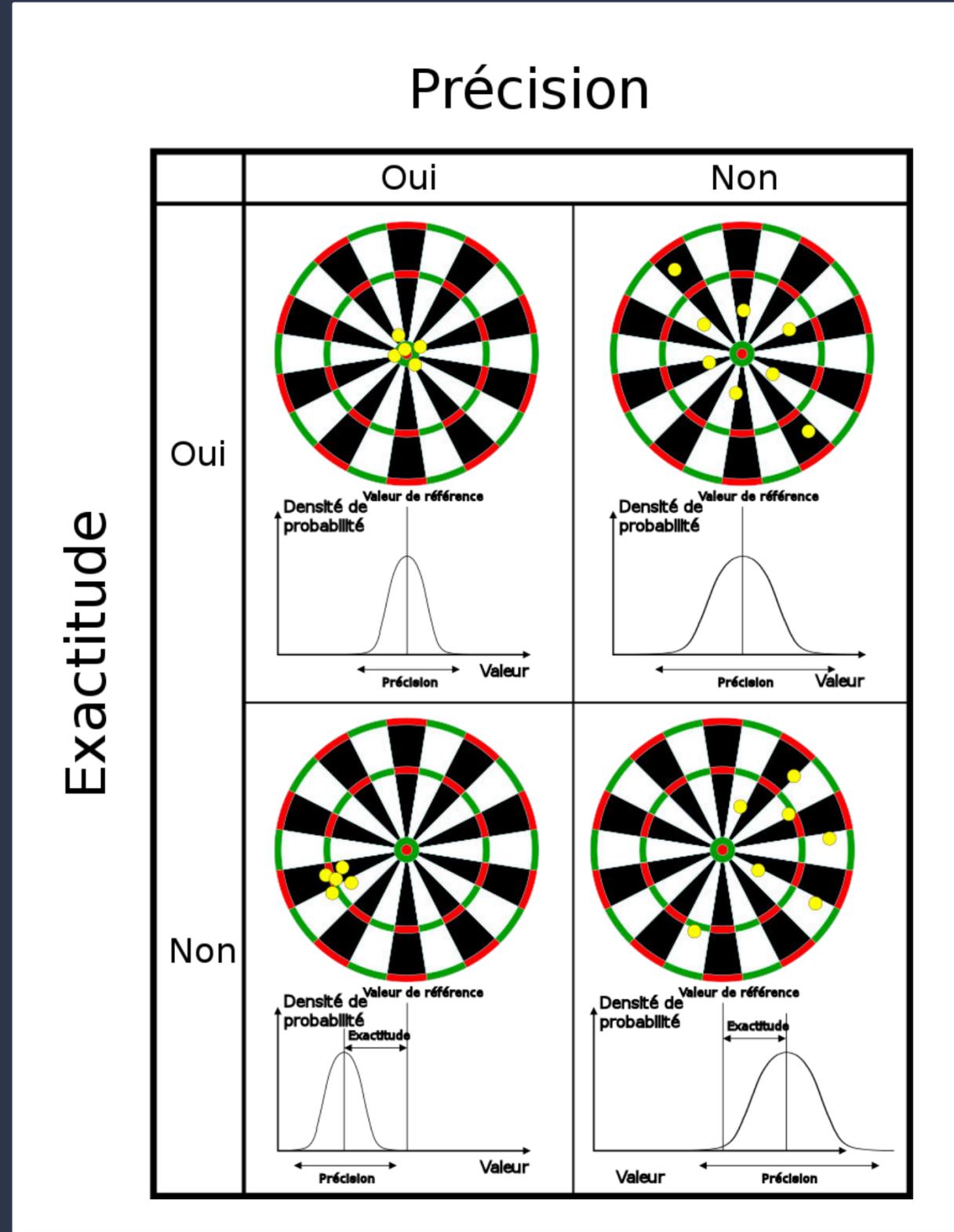


Importances des métriques mesurées
(Le nombre de requêtes ou le poids d'une solution n'est par exemple pas idéal)

- Mesurer des environnements potentiellement instables
- Analyser des métriques diverses



Compétence en métrologie à acquérir



Plus d'agilité, plus de test



un impact plus important de la chaîne de test

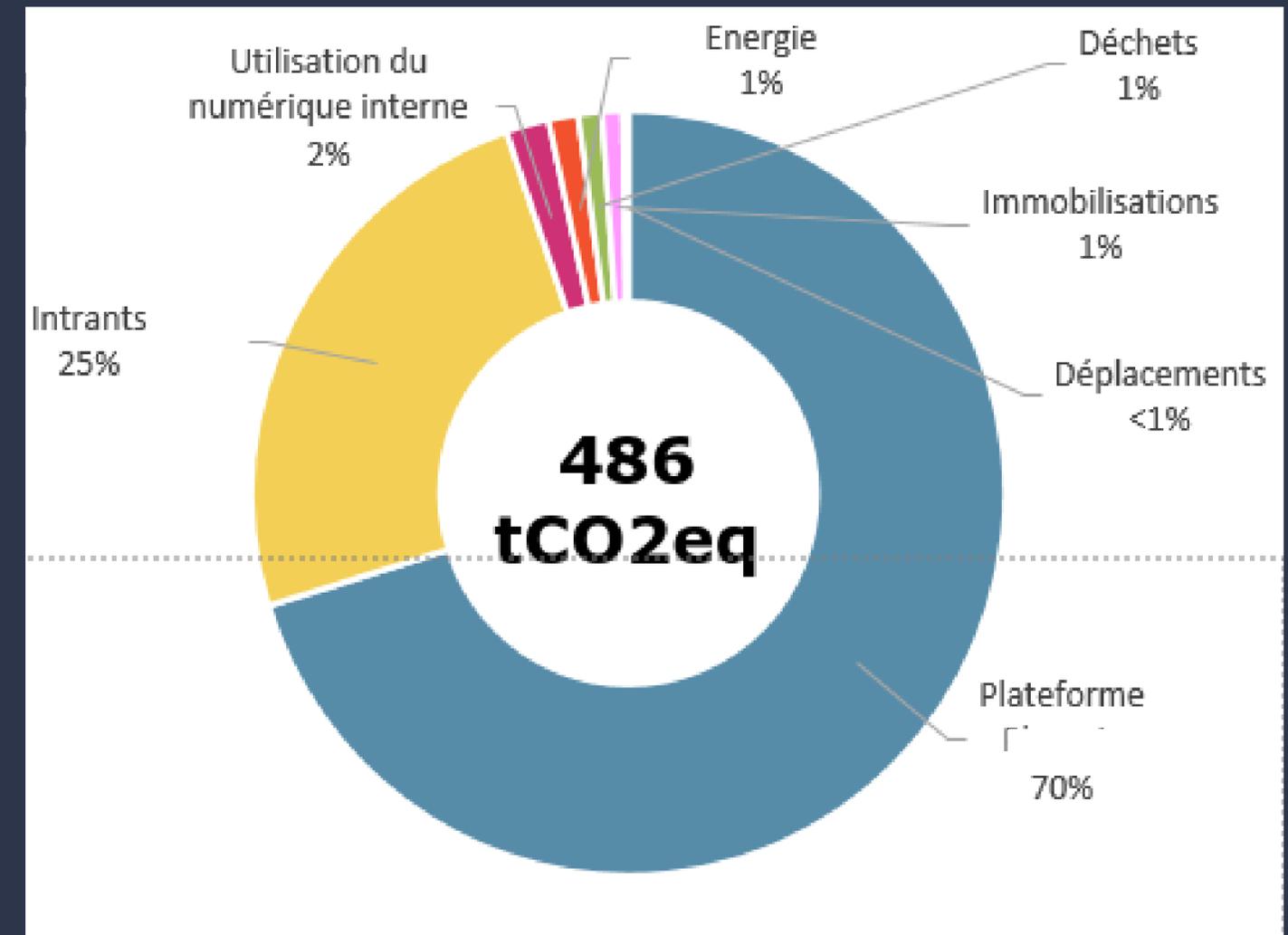


Réduire l'impact de la chaîne de test

Evaluation de l'impact plateforme audiovisuelle vs impact équipe

Impact de l'équipe plus faible mais non négligeable dès la 1 ère année

Nécessité d'appliquer des bonnes pratiques Green IT: Déplacement mais aussi extinction des environnements de tests...



Les challenges du DevGreenOps pour les équipes de test

Le Green dans la chaîne DevOps elle-même

Une professionnalisation et une structuration des outils de mesure

Plus d'automatisation

Une culture de la sobriété à intégrer

Olivier

Philippot

DevGreenOps : Opportunité de repenser la culture du test pour plus d'agilité et de collaboration

MERCI DE VOTRE ÉCOUTE
N'oubliez pas de voter

JOURNÉE
FRANÇAISE
DES TESTS
LOGICIELS