



# Tests de Performance, Agile et DevOps

Séminaire

Hello, we are valtech\_

Nous créons de la **Valeur** par la **Technologie**.

# Nouveaux contextes pour le test de performance



- **Déploiement continu**
  - Petits volumes, moins d'enjeu, mais plus de déploiements
  - Durée des tests de qualification
- **Mutualisation** des ressources
  - Impacts croisés
- **Cloud**
  - Capacité infinie, mais tarification à la consommation

# Stratégies d'évitement à éviter

- Le test est censé garantir les performances en production
  - Plate-forme Iso-Prod
  - Mix de scénarios représentatifs
- Pas le temps, on fonce ...
- Test Canada Dry



# Transformation de l'objectif

- Le test est censé garantir les performances en production
  - Plate-forme Iso-Prod
  - Mix de scénarios représentatifs



- Tenir SLA en production
- Préparer la production
  - Avoir une application saine
  - Valider le monitoring
  - Comprendre les points de fragilité
  - Optimiser pour réduire les consommations de ressources

# Définir les risques de performance

- disponibilité critique
- temps de réponse critique
- volume des données important
- nombre d'utilisateurs très élevé
- traitement intensif



# Design for failure

- Les tests n'éliminent pas tous les risques
- **Se préparer** pour les aléas de production
  - Architectures flexibles
  - Modularité
  - Résilience
  - Monitoring et logs
  - Gestion des anomalies





# Tester au plus tôt en night build

- Limiter les besoins de représentativité



capacité

- gérer par l'élasticité de la plate-forme
- identifier les fuites à faible charge

- identifier les hot-spots isolément et à faible charge

instabilité

algorithmes

contentions





# Améliorer la représentativité des tests : Dark Launching

- Livraison de fonctions **non visibles** de l'utilisateur **mais utilisées** à son insu



- Par exemple duplication des requêtes sur la nouvelle version
  - pas d'impact si la nouvelle version échoue
  - permet de collecter de l'information

# Réduire le risque : Soft Launching

- Démarrage graduel
- Activer les features pour **une partie des demandes**
  - Par configuration de l'**instance**
  - Sur **demande de l'utilisateur**
  - Selon l'**utilisateur**
- Feature Flags
- Attention au partage de la base pendant la période de transition



# Chaos Monkey

- <https://github.com/Netflix/SimianArmy>
- Equivalent des tests d'intrusion dans la sécurité
- Teste la capacité à survivre à différent types de problèmes
  - Résilience
  - Opérations



---

# Conclusion

---

- Utiliser les tests pour **déminer** et comprendre à quoi la plateforme est **sensible**
- **Limiter** les tests représentatifs et iso-production.
- Envisager le **Dark Launching**
- Gérer la capacité via l'**élasticité** des plates-formes
- Réduire les risques lors des lancements (**Soft Launching**)
- **Monitorer**
- **Tester le monitoring** et la résilience



**valtech\_**



**valtech\_**