



Equipes Agiles & DevOps : Testez la valeur d'abord !



LAURENT PY

@py_laurent

laurent.py@hiptest.net

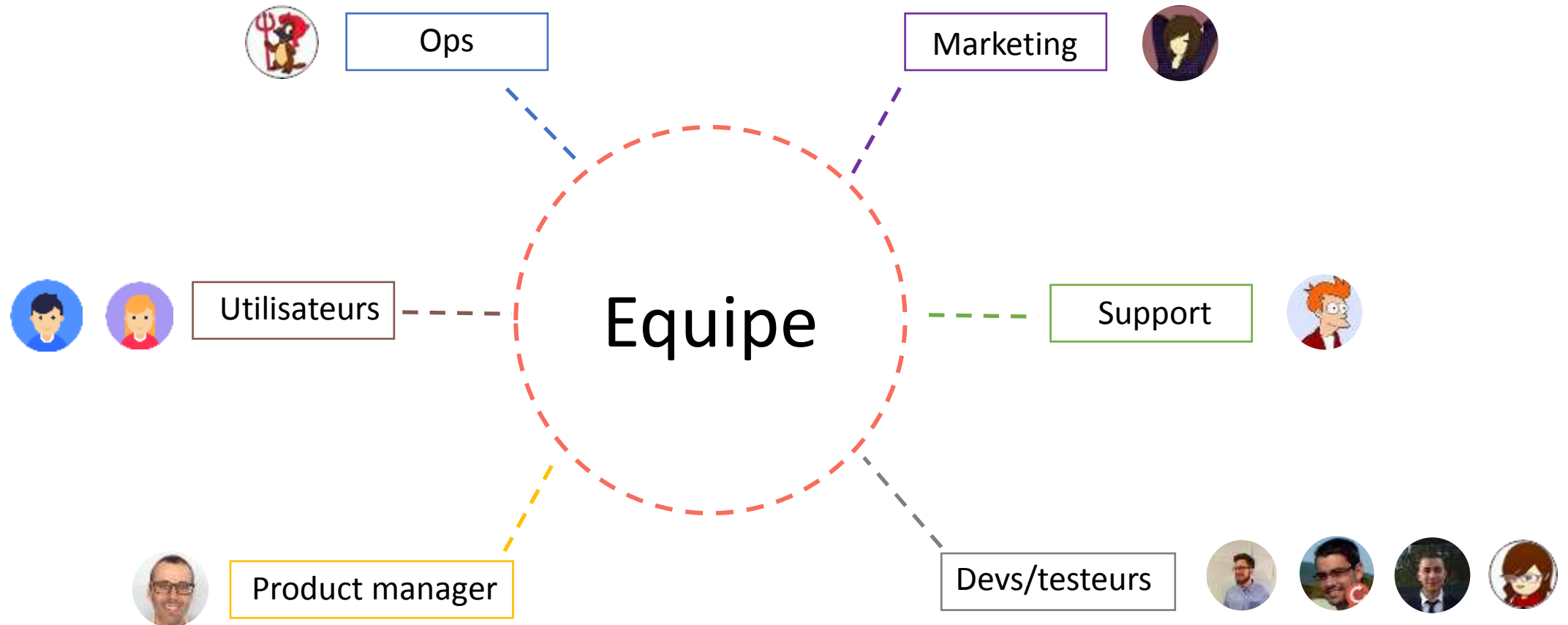
<https://hiptest.net>

PLAN

- Le match vitesse VS qualité
- Étape 1 : Tester l'idée
- Étape 2 : Premiers tests en production
- Étape 3 : Consolidation des tests
- Conclusion : Bénéfices

VITESSE VS QUALITÉ

NOTRE EQUIPE DEVOPS

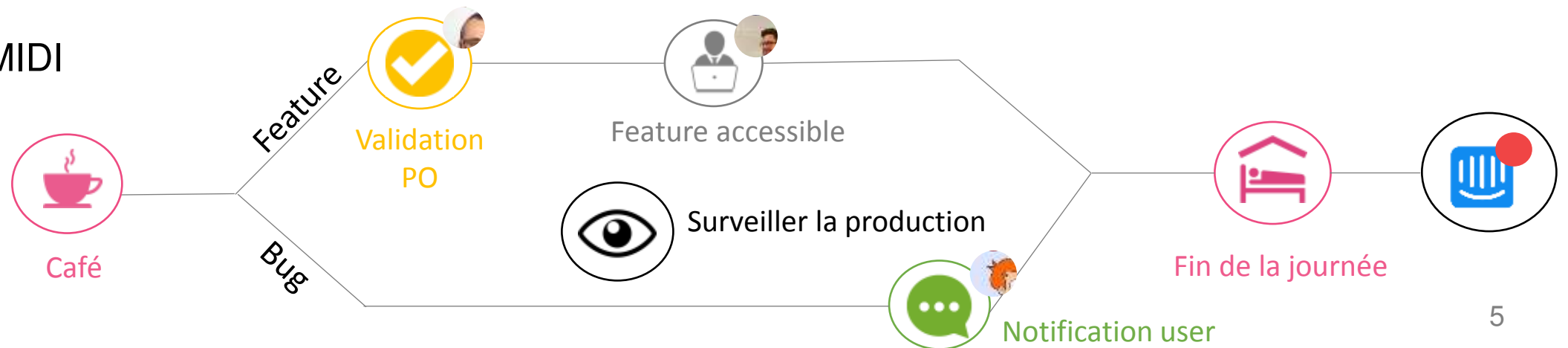


UNE JOURNÉE AVEC UNE ÉQUIPE DEVOPS

MATIN



APRÈS-MIDI



RETOUR DE L'UTILISATEUR SUR LE LIVE CHAT



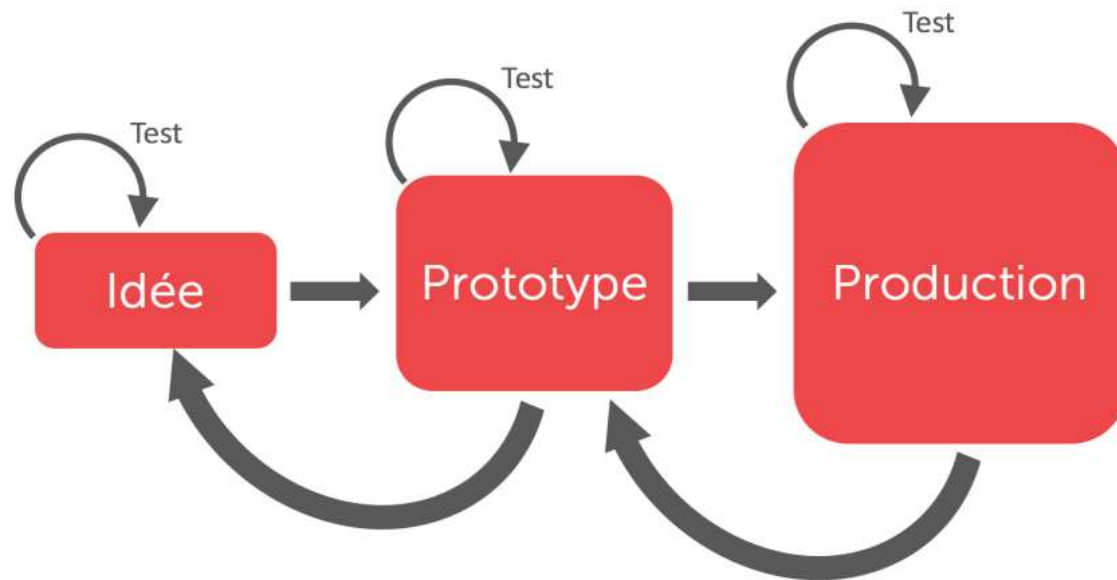
"I like what I've seen so far and I've been very impressed with the live support. Your team managed to identify and fix a bug I'd found within about 12 hours."

Tom Graham
Product Innovation Software Manager



ET LE CLIENT/UTILISATEUR DANS TOUT CELA?

L'enjeu n'est pas la vitesse de développement, c'est la rapidité d'apprentissage et la flexibilité



Valeur VS Correct

DÉVELOPPEMENT D'UNE FONCTIONNALITÉ

Nous avons structuré notre processus de test autour des 3 questions suivantes:

- 1) Quelle valeur voulons-nous livrer à l'utilisateur ?
- 2) La fonctionnalité a-t-elle réellement un impact ?
- 3) Quelle est la performance du service et la fonctionnalité est-elle toujours utilisée dans le temps ?

ÉTAPE 1: TESTER L'IDÉE

QUELLE VALEUR VOULONS NOUS LIVRER?

- Utilisation de personae pour créer une première “intimité” client
- Définir des hypothèses business
- Behavior Driven Development (BDD) pour capturer la discussion

The diagram illustrates the application of Behavior Driven Development (BDD) to create user personas and capture requirements. On the left, five personas are listed: Devs/testeurs, Product manager, Marketing, Support, and Ops. On the right, a BDD scenario is shown in a light gray box. A red arrow points from the text "Bénéfice attendu" (Expected benefit) to the scenario's goal. The scenario text is: "In Order to stay up to date directly within my messaging App As a team member I want Hiptest to notify changes on my messaging App". The box also includes metadata: "Stay up to date with Slack integration", "Created by Laurent Py at 21 Nov 21:18", "Updated by Laurent Py at 22 Nov 21:59 (Updated name)", and a "Revision history" dropdown showing 0 and 3 revisions.

Product manager

Devs/testeurs

Marketing

Support

Ops

Stay up to date with Slack integration

Created by Laurent Py at 21 Nov 21:18.
Updated by Laurent Py at 22 Nov 21:59 (Updated name)

In Order to stay up to date directly within my messaging App
As a team member
I want Hiptest to notify changes on my messaging App

Bénéfice attendu

Revision history

#BDD in a tweet: Using examples at multiple levels to create a shared understanding and surface uncertainty to deliver software that matters



BDD EN UN CLIN D'OEIL

- Une compréhension partagée du système avec des exemples
- Basé sur un langage métier commun
- Définition du stop
- Lorsque que les tests/exemples sont automatisés, ils deviennent la documentation vivante.

CAPTURE DU COMPORTEMENT ATTENDU

Stay up to date with Slack integration

Created by Laurent Py at 21 Nov 21:18.
Updated by Laurent Py at 22 Nov 21:59 (Updated name).

In Order to stay up to date directly within my messaging App
As a team member
I want Hiptest to notify changes on my messaging App

0 3
Revision history ▾

Création d'un ou plusieurs scénarios Gherkin

- 1 **Given** I have activated Slack in the settings of my Hiptest project
- 2 **When** I query /hiptest TestRunID from slack
- 3 **Then** I should see the breakdown of tests by statuses for TestRunID

Add step



Parfois interview client



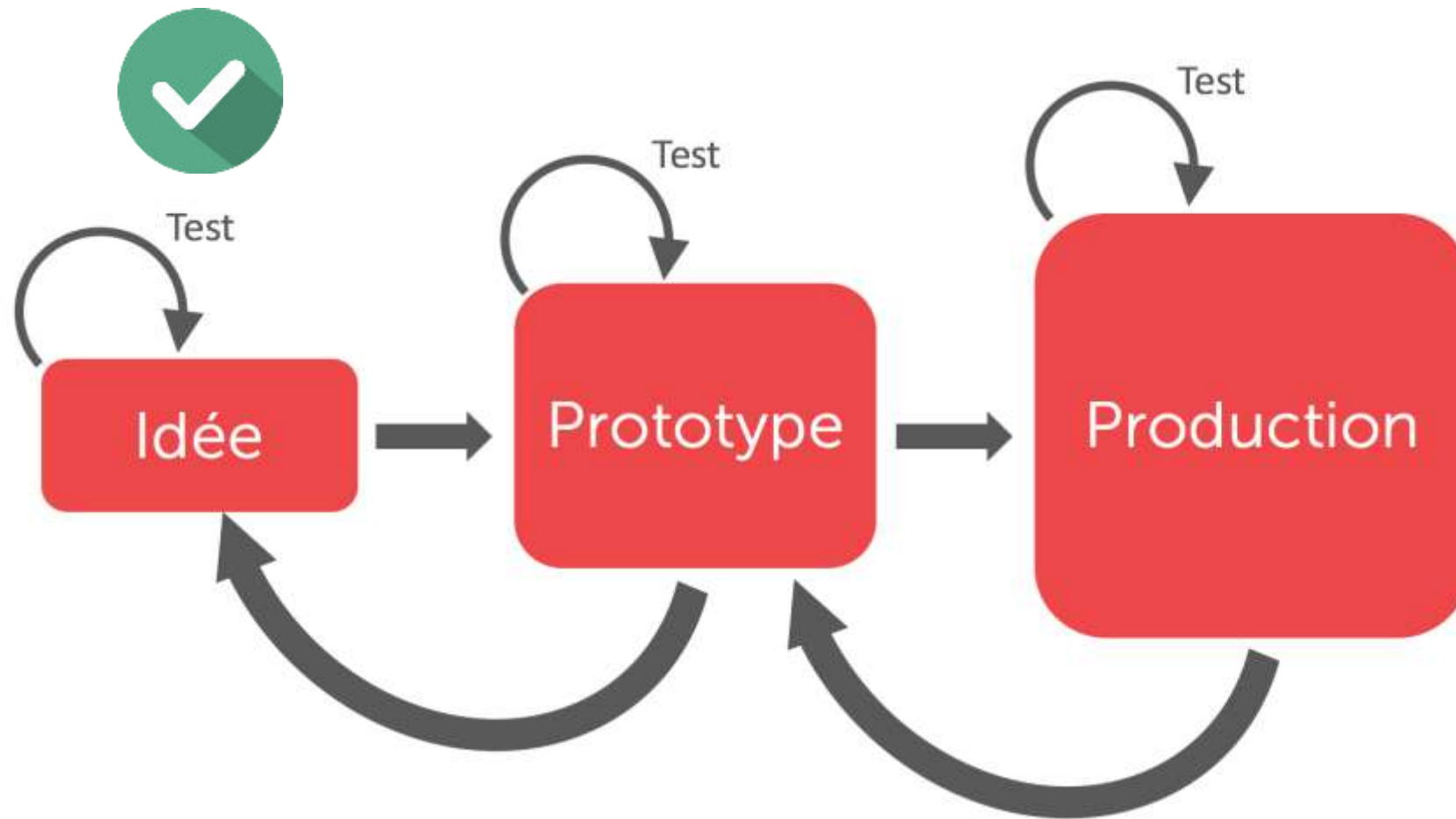
QUELQUES BONNES PRATIQUES

- Utiliser la syntaxe Gherkin (style déclaratif)
- Etre cohérent dans la terminologie métier utilisée

```
1 Given I have activated Slack in the settings of my Hiptest project Coffee Machine
2 When I query /hiptest TestRunID from slack
3 Then I should see the breakdown of tests by statuses for TestRunID

and I should|
Create action word
I should
"user" should be logged in (called 2 times)
I should be member of project "p1" (called once)
I should get a notification "p1" (called 8 times)
```

- Refactoring continu des tests



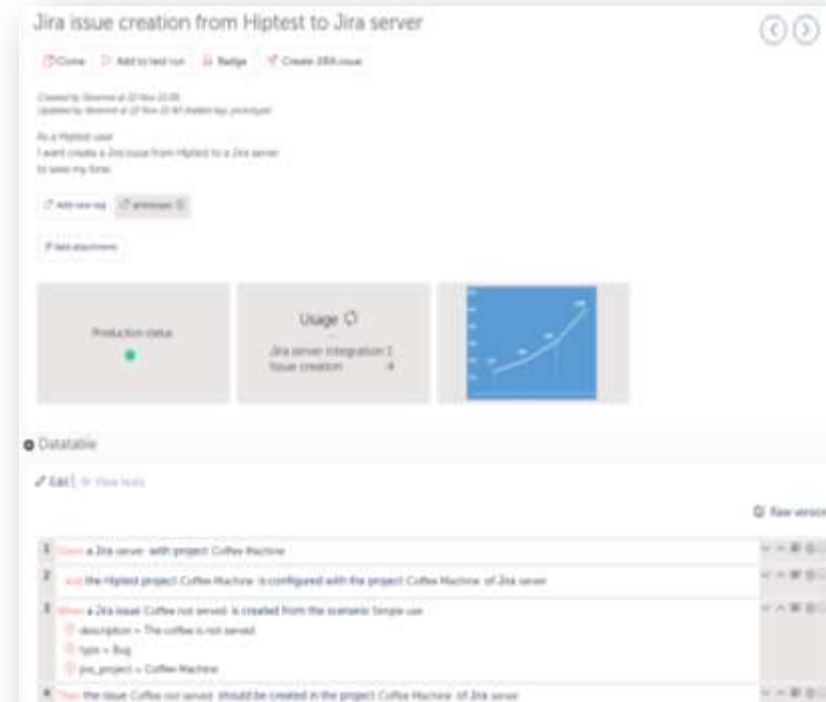
ETAPE 2: PREMIERS TESTS EN PRODUCTION

DÉVELOPPEMENT

1. L'étape 1 a permis de dégager une compréhension partagée de la feature (comportement & impact)
2. L'équipe développe un premier niveau fonctionnel guidé par les scénarios
3. Déploiement en production (limité ou non)

MESURE DE L'IMPACT

- Mesure de l'usage
 - NB users
 - NB requêtes...
- Mesure de l'impact
 - Sign-up
 - Revenu
 - Rétention...



La discussions sur les indicateurs consolide la compréhension partagée !

EXEMPLES DE MESURE

Fonctionnalité de Notifications

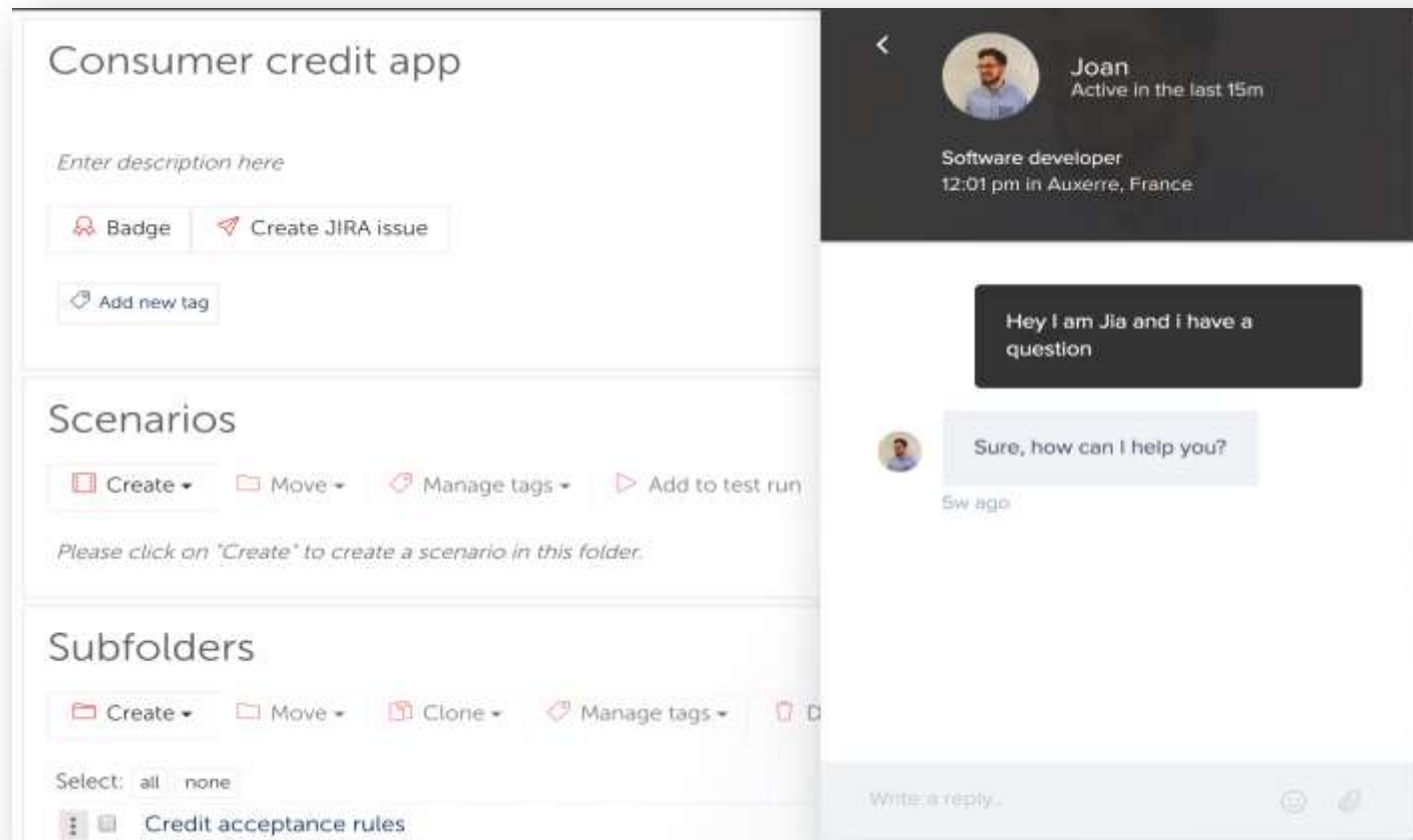
- Usage
 - NB clicks
 - NB de notifications
 - NB utilisateurs
- Business: aucune

Fonctionnalité de Refactoring

- Usage
 - NB de refactoring
 - NB d'impacts
- Business: rétention



LIVE CHAT POUR FEEDBACK TEMPS RÉEL

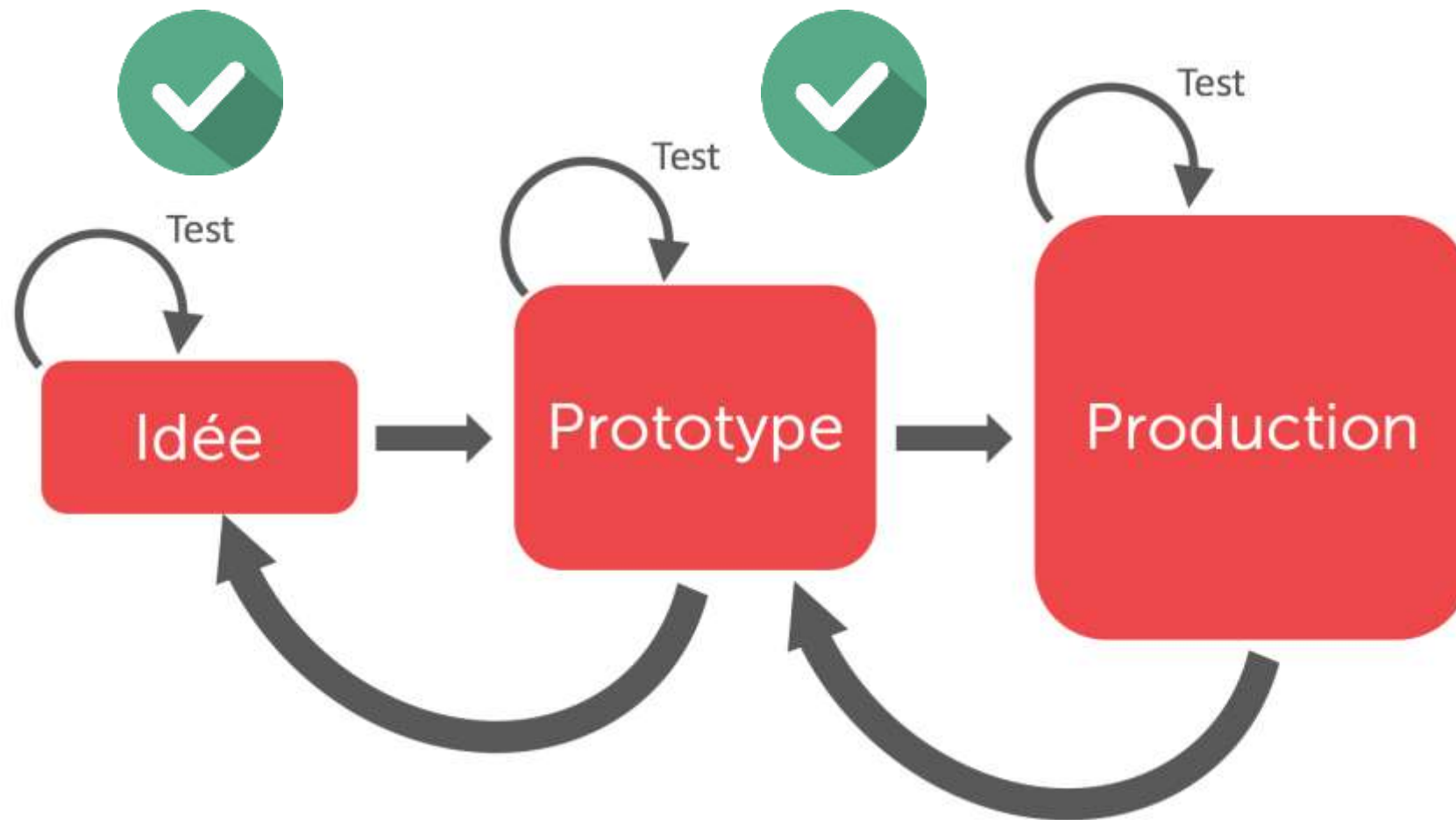


EAT YOUR OWN DOG FOOD

- L'équipe utilise son propre produit pour le test
- Autre exemple : refonte de l'application envoi de colis par TNT



L'équipe est allée physiquement envoyer des colis et a trouvé des bugs.

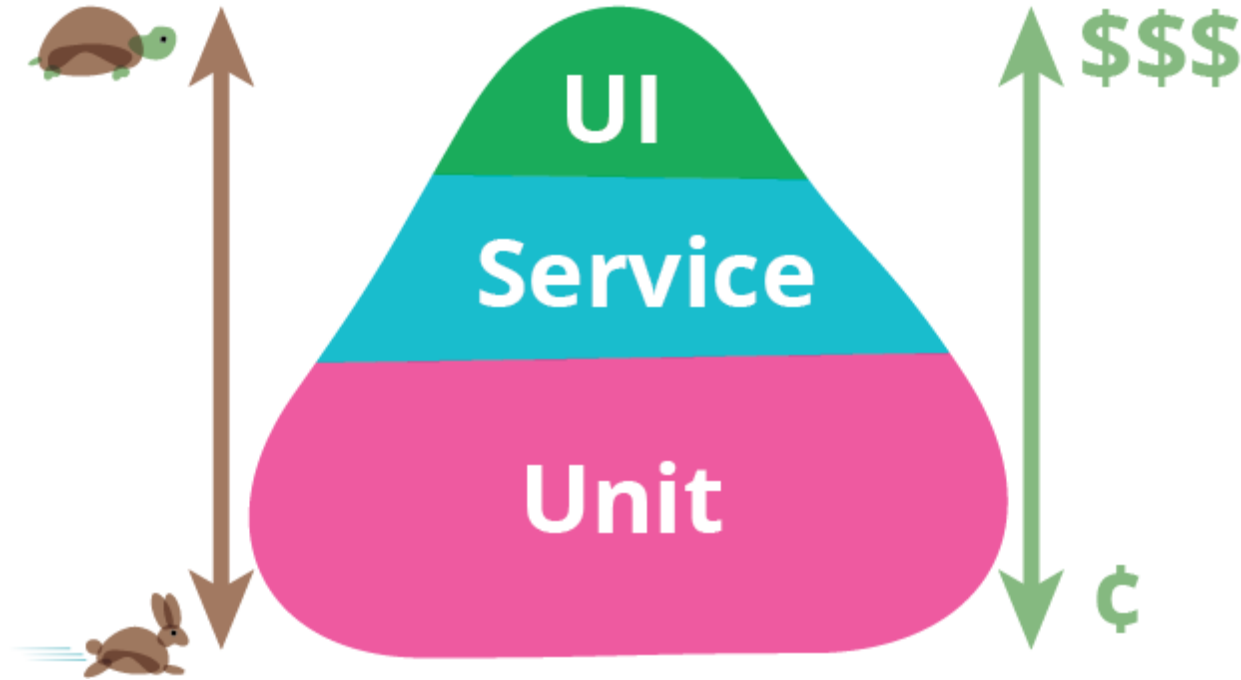


ETAPE 3: CONSOLIDATION DES TESTS

MISE À DISPOSITION GÉNÉRALE

1. L'étape 2 a permis de valider l'impact sur les utilisateurs
2. Investissement sur la performance et les tests de non régression (automatisation 100%)
3. La fonctionnalité est généralisée à l'aide d'un switch applicatif

NOS TESTS AUTOMATISÉS



Credit to Martin Fowler

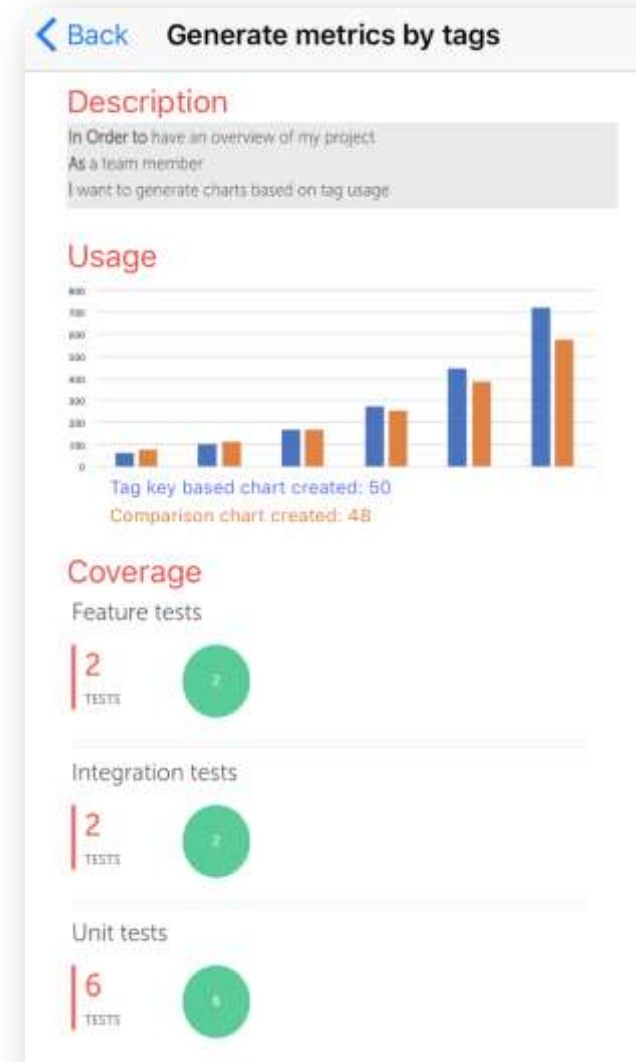
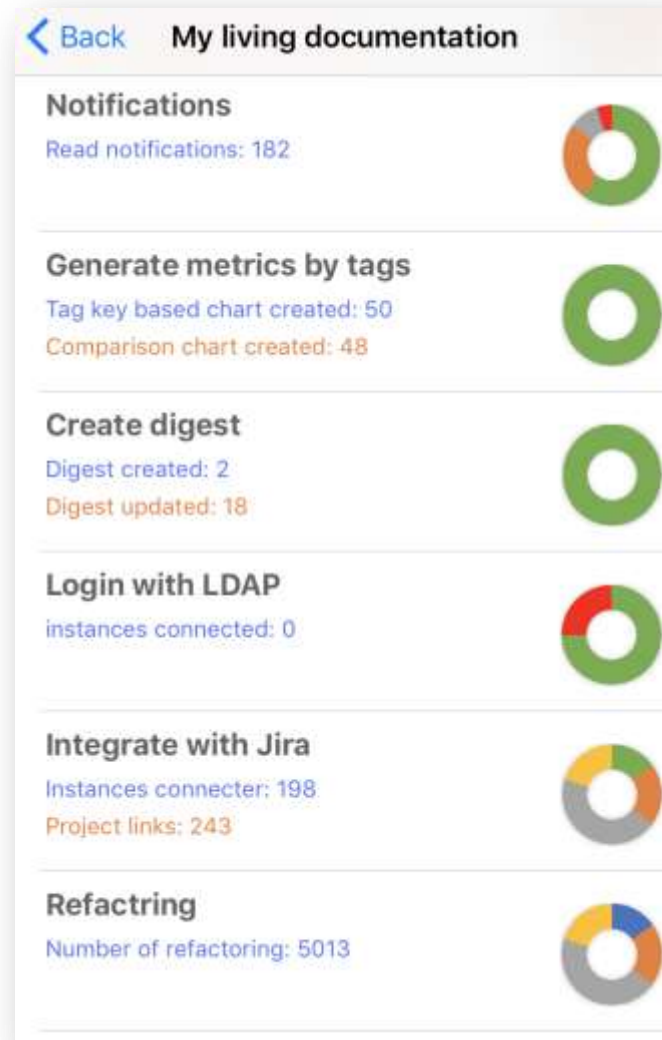
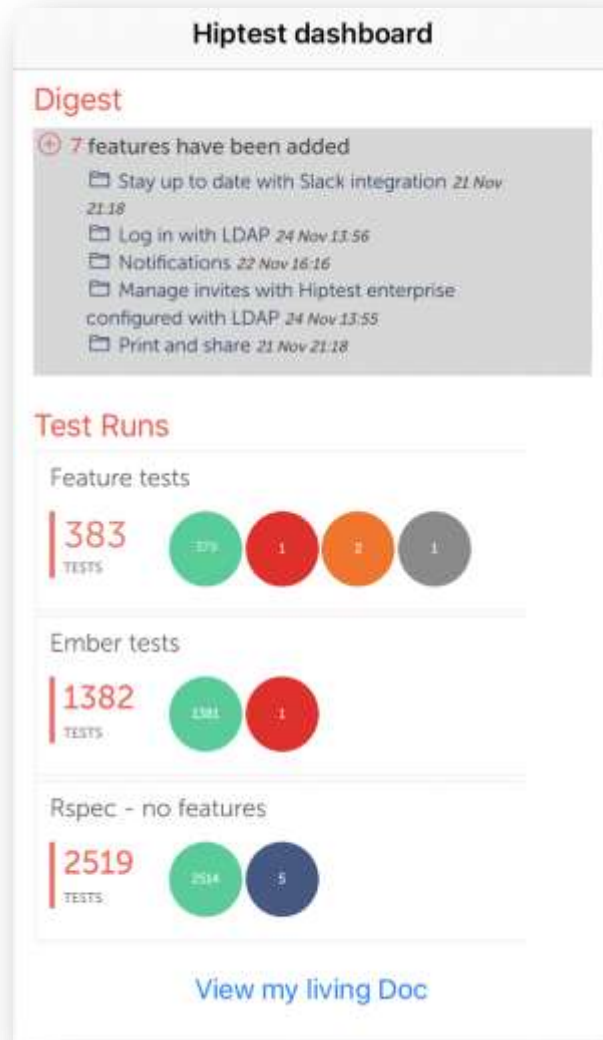
- 450 tests BDD
- 1400 tests Ember
- 2600 tests Rspec

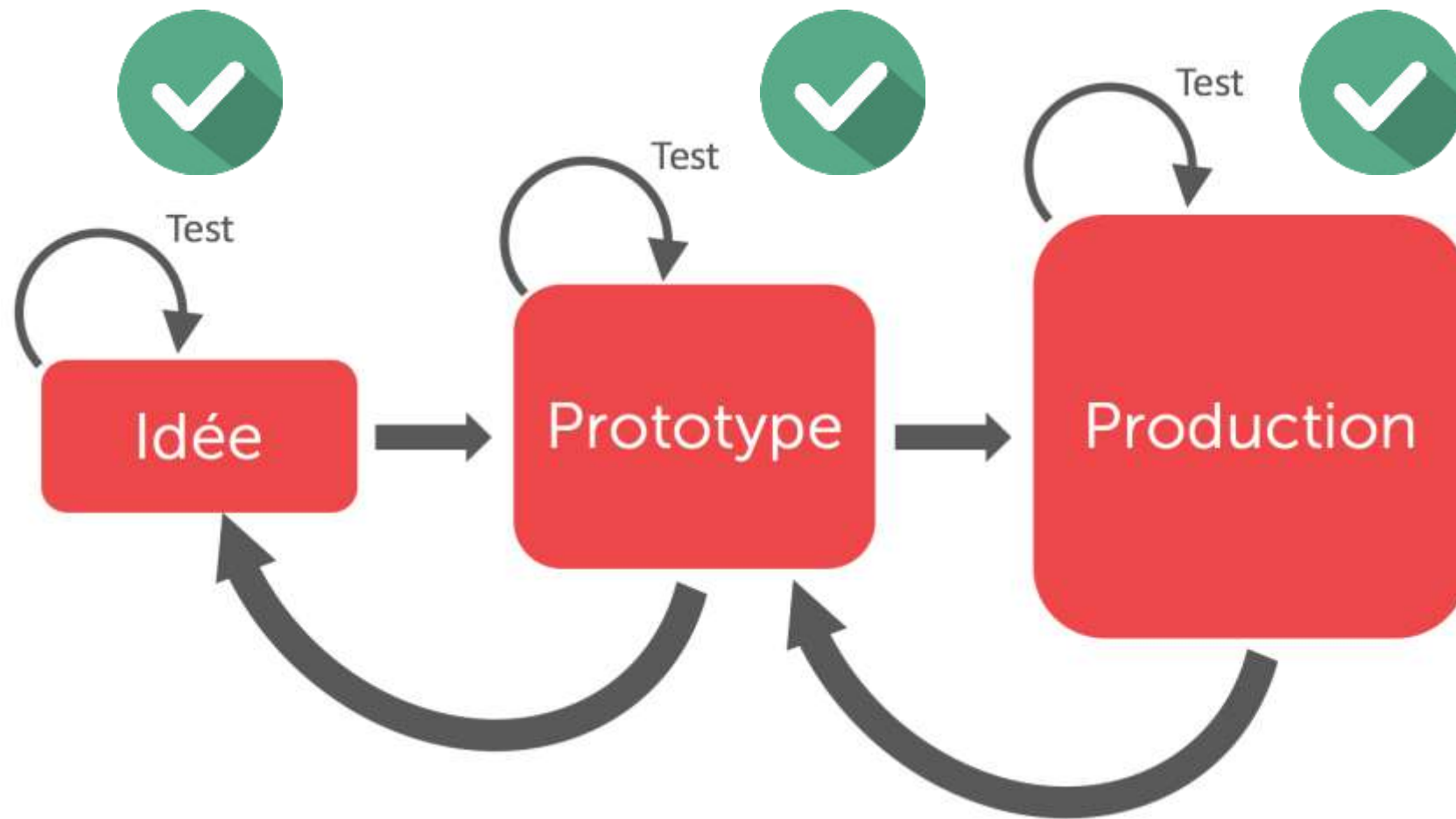
MONITORING DES PERFORMANCES EN CONTINU



Appsignal, Pingdom, Scalingo et Logmatic

MONITORING DE L'USAGE EN CONTINU





CONCLUSION

BÉNÉFICES DE L'APPROCHE

- Tester la valeur au plus tôt et en continu
- Investissement progressif sur la qualité
- Porteur de sens pour l'équipe au quotidien

TOUS RESPONSABLES

- Pas de stress au déploiement
- Environnement propice aux **expérimentations** et changements
- Bon fonctionnement du service porté par tous