

Agile + DevOps **EAST**

A TECHWELL EVENT

DT9

Integrating Testing into DevOps

Thursday, November 8th, 2018 1:30 PM

A DevOps Team's Journey Toward Behavior-Driven Development

Presented by:

Laurent Py and Vincent Prêtre

Hiptest

Brought to you by:



350 Corporate Way, Suite 400, Orange Park, FL 32073
888-268-8770 · 904-278-0524 - info@techwell.com - <http://www.starwest.techwell.com/>

Laurent Py

With more than fifteen years of experience as a developer and product manager, Laurent Py is now product manager at Hiptest, a continuous testing platform dedicated to agile and DevOps teams. Hiptest helps product owners, developers, and testers collaborate to build the right product faster.

Vincent Prêtre

Vincent Prêtre is a cofounder of Hiptest with more than ten years of software experience. He is specialized in web development and is also the creator and main contributor of the open source framework Hiptest-Publisher. He has been a speaker at several conferences around the world to share his experience about testing and behavior-driven development.



The journey of our DevOps team to Behavior Driven Development

Presented by



Laurent Py

@py_laurent
laurent@hiptest.com
<https://hiptest.com>



Vincent Pretre

vincent@hiptest.com
<https://hiptest.com>

Agenda

- Testing in DevOps
- Step 1: Testing the idea
- Step 2: Testing the code
- Step 3 : Testing in production
- Step 4 : Let's iterate
- Conclusion

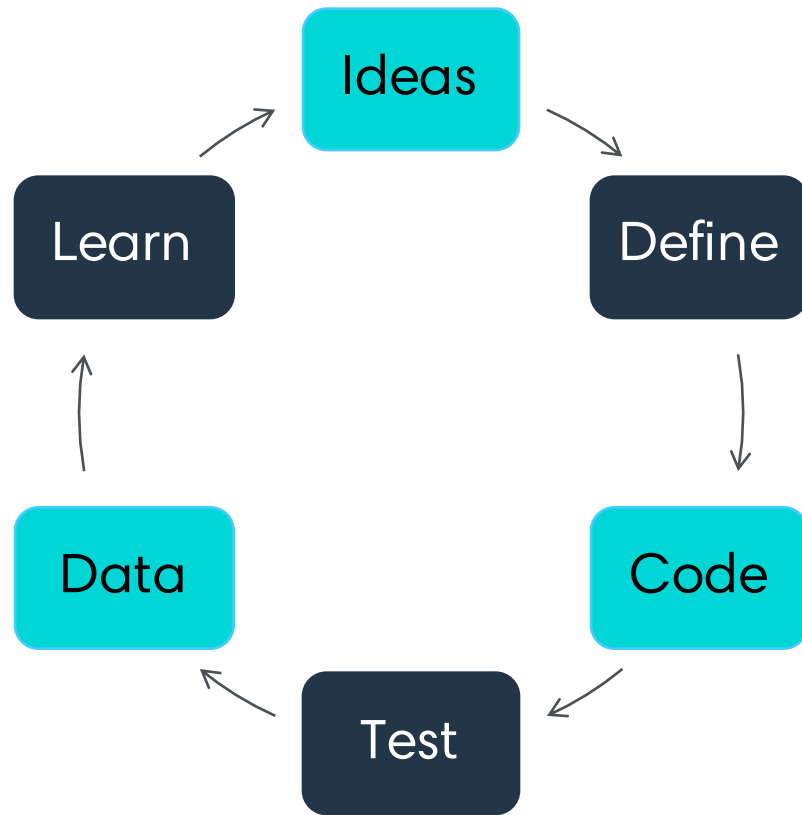
Testing in DevOps

DevOps: Biggest opportunity?



SPEED!

DevOps: Biggest opportunity?



Minimize total time through the loop.

Testing in DevOps

1. Which value do we want to deliver?
2. Is the feature's implementation aligned with that?
3. Does the feature have a real impact on the end user?
4. Is the UX great and what's the feature usage in time ?

Step 1 : Testing the idea

Why?

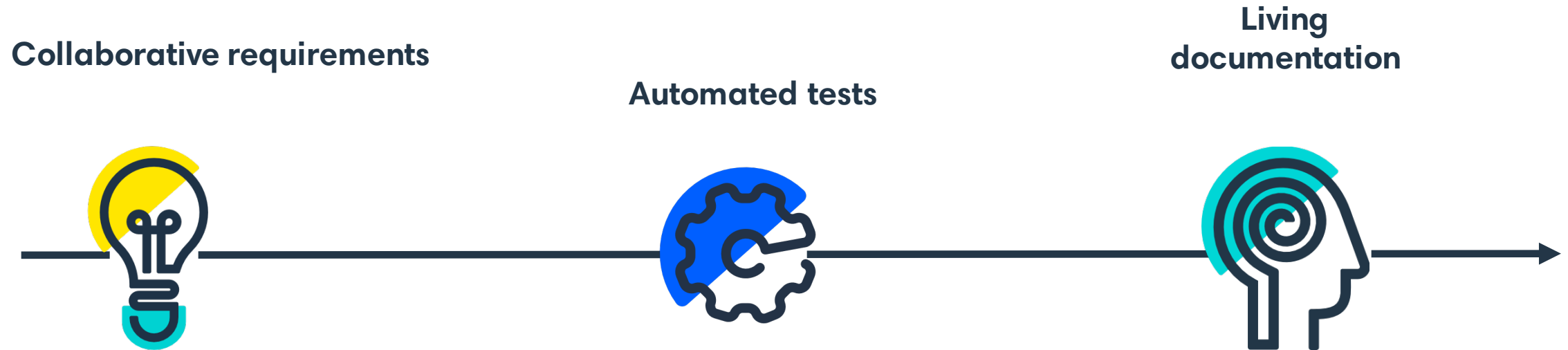


Challenge business assumptions

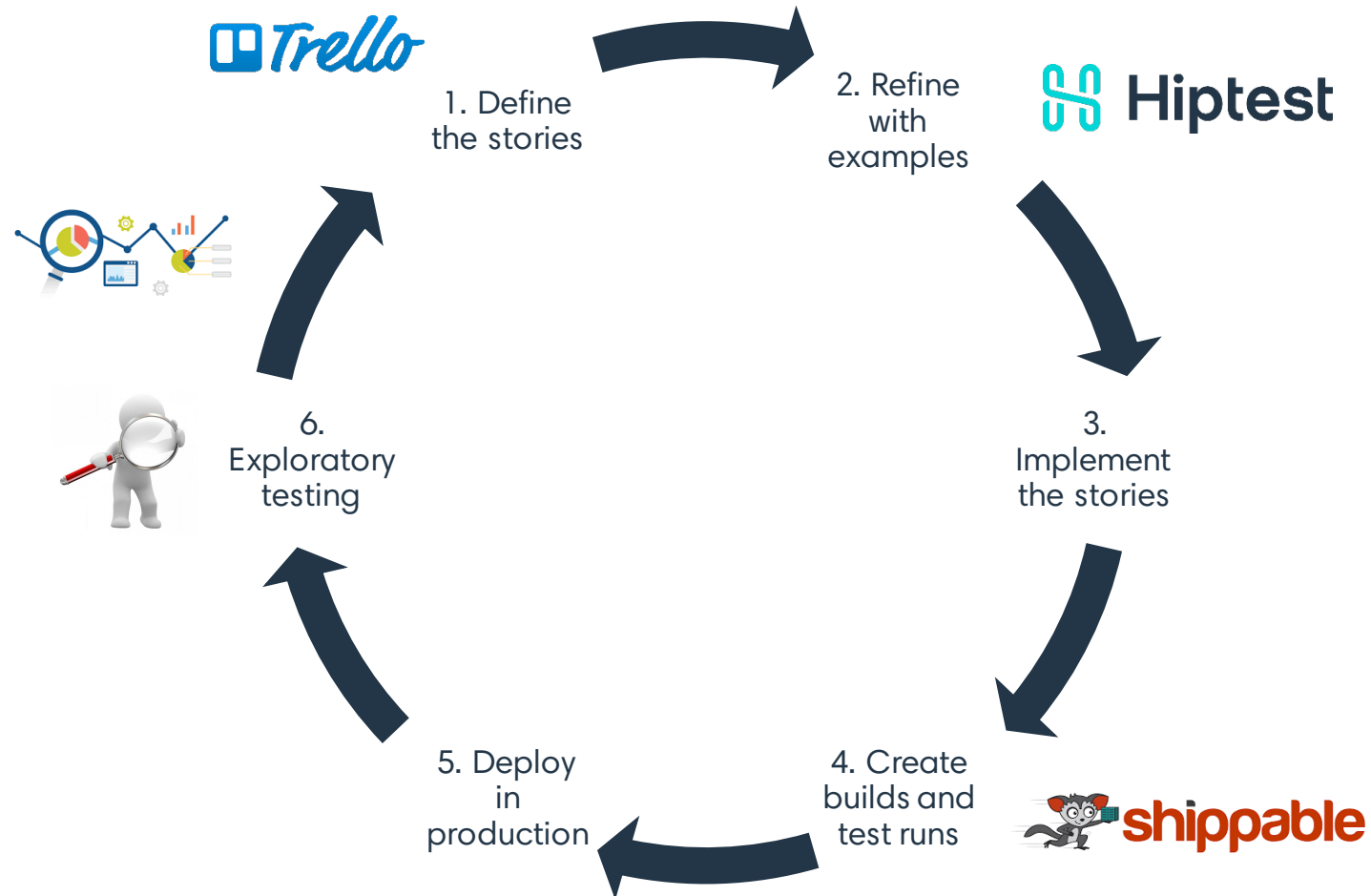
Which value do we want to deliver?

- Use Personae/JTB to create intimacy between users and the team
- Define business assumptions
- Behavior Driven Development (BDD) to capture the behavior

What's BDD?



Testing process at Hiptest



BDD – Capture the behavior

The screenshot displays a BDD tool interface with a dark sidebar on the left. The sidebar contains a logo, a search icon, a plus icon, and several sections: **DEFINE** (Scenarios, Action words, Metrics), **TEST** (Test runs, Automation), **LEARN** (Living documentation), and Project settings. The main area is divided into three panes. The top pane shows a search bar and a list of features under the heading 'Coffee machine - Hiptest pu...'. The middle pane shows the details of the 'Beans' feature, including its background and messages. The bottom pane shows the 'Beans' feature content, including its background and messages.

Beans

As a coffee lover
I have to put fresh beans from time to time
So I can have coffee when I need it

Background

Given the coffee machine has been started
And I handle everything except the beans

Message "Fill beans" is displayed after 38 coffees are taken

priority: high

When I take 38 coffees
Then message Fill beans should be displayed

It is possible to take 40 coffees before there is really no more beans

priority: low

When I take 40 coffees

Attempt #1

Given user Harry

and project with errors

and a test run named TR1

When test unknown var is selected

Then test has error Interpretation error: unknown symbol 'v'

When test datatable not totally full - Ko is selected

Then test contains use

When test datatable not totally full - Ok is selected

Then test contains use value

When test syntax error is selected

Then test has error Interpretation error: syntax error

Attempt #2

Generate metrics by tags

Export ▼

...

In Order to have an overview of my project
As a team member
I want to generate charts based on tag usage

 Add attachment

Add new tag



2 scenarios

Create

All / None Show tags

A chart can be generated based on a tag key

A chart can be generated to compare multiple tags

Attempt #3

Given all test results are passed

and no execution cycle is started

When I start a new execution cycle

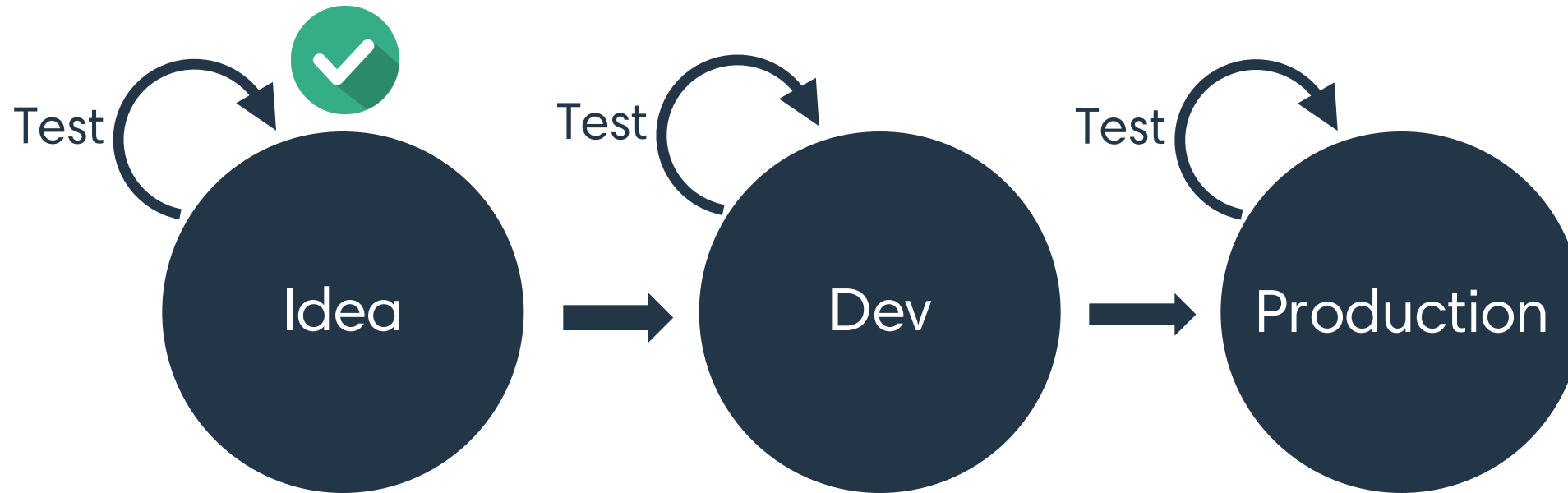
Then the test run Sprint 1 should contain a started execution cycle

and all test results should be not run

Good practices

- Use of gherkin syntax on purpose (Given / When / Then)
- Use of **declarative style**
- Reuse a consistent **business terminology**
- Review and **refactor** your BDD scenarios continuously

Continuous testing



Test continuously your software from idea to production

Step 2 : Testing the code

Development

1. In step 1 we have created a shared understanding of the feature
2. The dev' team can develop a first increment of the feature
3. Integration to CI/CD pipeline

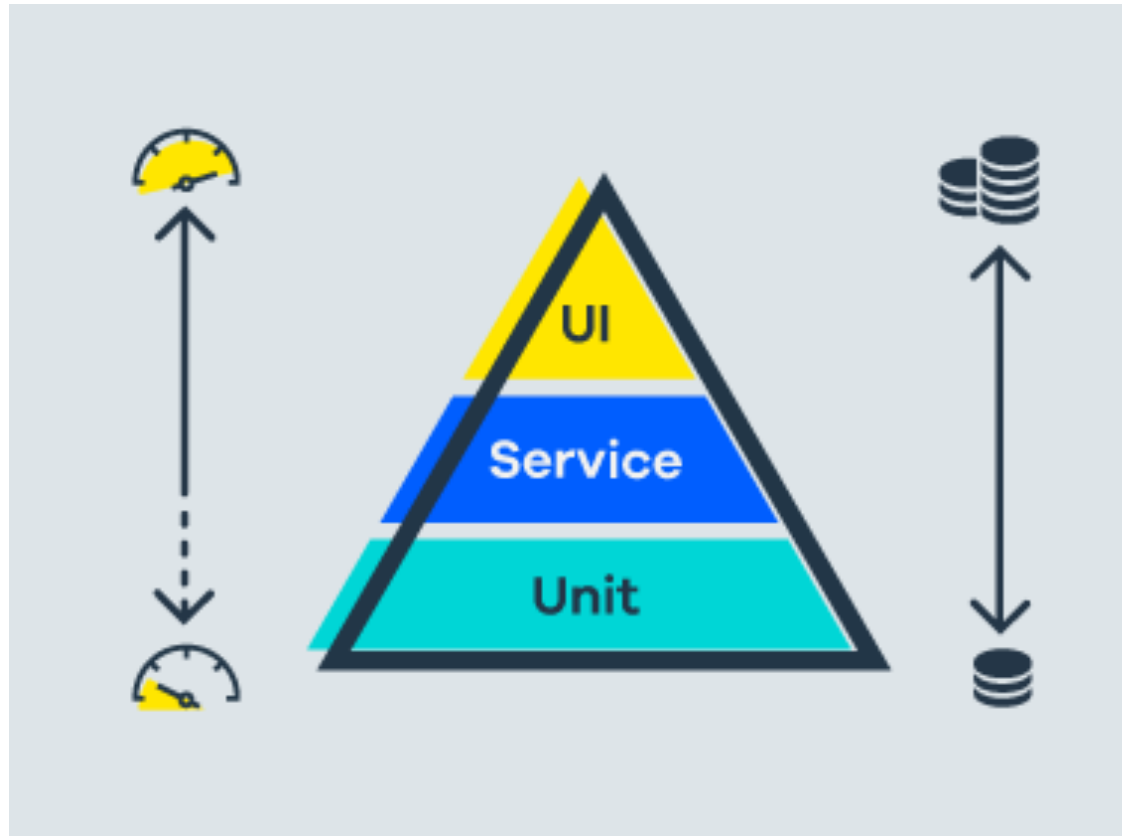
Automation

100% of our checks (scenarios) are automated

And, it's **expensive!**



Our automated tests



- 500 tests BDD
- 1250 tests Ember
- 4000 tests Rspec

Right level of automation

Given all test results are passed

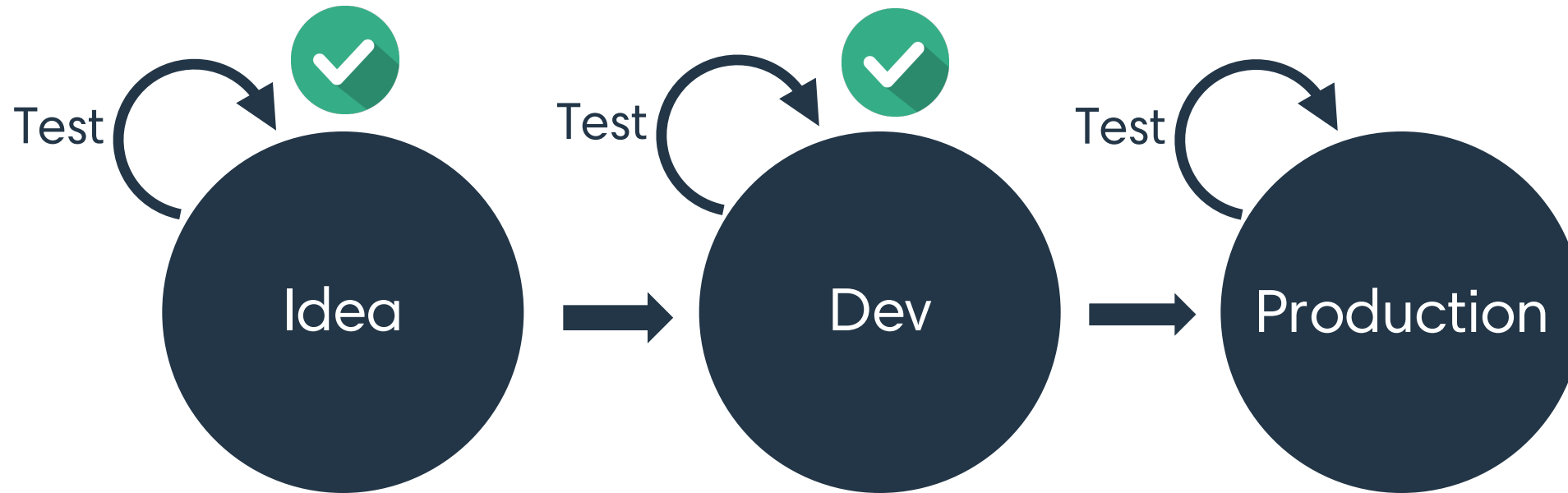
and no execution cycle is started

When I start a new execution cycle

Then the test run Sprint 1 should contain a started execution cycle

and all test results should be not run

Continuous testing



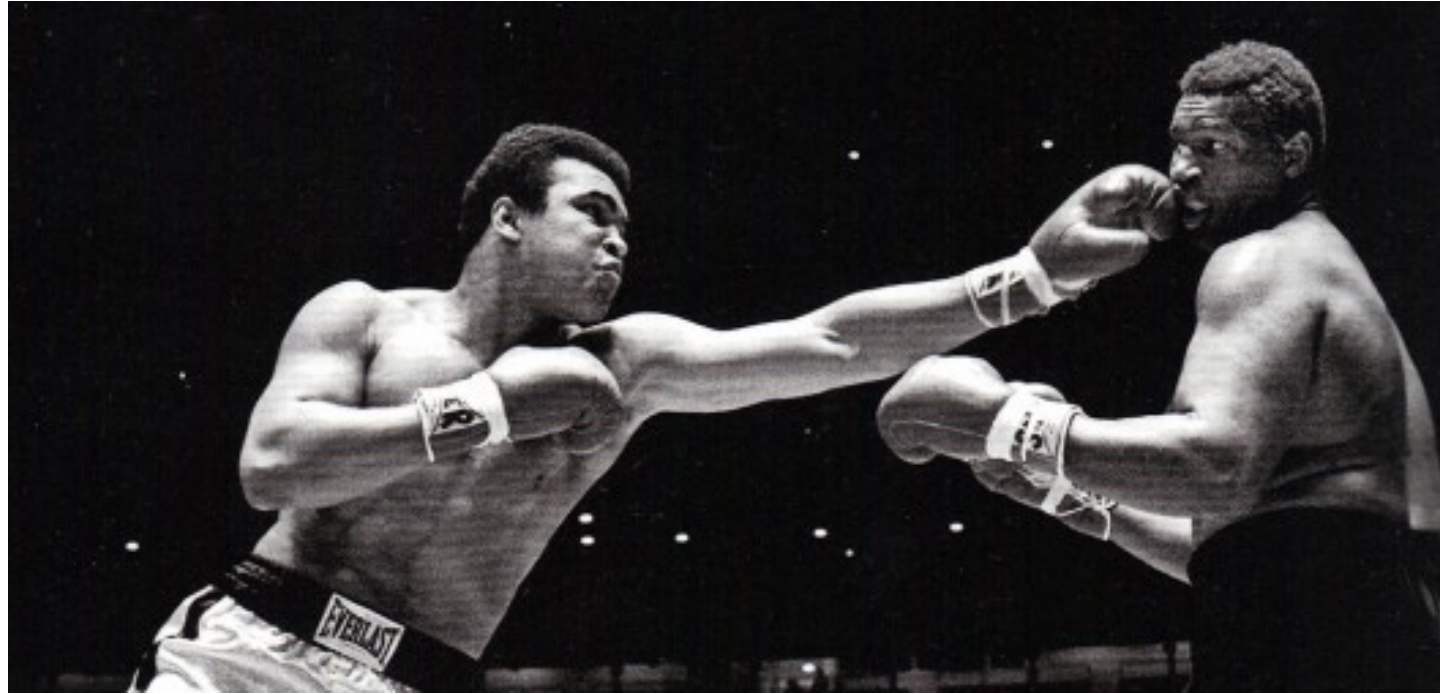
Test continuously your software from idea to production

Step 3 : Testing in production

Making the feature available

1. In Step 2 we have validated the implementation
2. The feature is rolled out using a switch (GA or progressive)

The match



Availability VS Correctness

Application performance management



Appsignal, Alertsight, Scalingo & Logmatic/Datadog

Living Documentation & Product analytics

The screenshot displays the Hiptest application interface. On the left is a dark sidebar with navigation options: Dashboard, DEFINE (Scenarios, Action words, Metrics), TEST (Test runs, Automation), LEARN (Living documentation), and Project settings. The main content area is titled 'CI - Reboot' and features a search bar and a tree view of features. The 'Share business terminology across projects' feature is selected, showing its business and technical approaches, business metrics, and background. A red oval highlights the business metrics section.

Share business terminology across projects

Business approach

When I test multiple close projects
I want to use the same business terminology
So I can ease the understanding of the examples for everyone

Technical approach

When I test projects that rely on the same technology
I want to reuse an existing action words
So I can speed up my scenarios writing and maintain my action words in a single place

BUSINESS METRICS

- Number of calls to shared action word
- Action word duplication rate within projects that belong to the same organization

Background

Given I am a valid Hiptest user
And I am administrator of the organization Hiptest
And the project Coffee machine belongs to organization Hiptest
And the actionword I take a coffee is shared within the organization
And the project Testing BOOKSTORE application belongs to organization Hiptest

Living Documentation & Product analytics

Test steps

- 1 **Given** I am a valid Hiptest user
- 2 **and** I open project My project
- 3 **and** I open living documentation for branch master
- 4 **When** I get the story of feature MyFeature
- 5 **Then** I should see all the feature updates since its creation

Create living documentation

Filters:

DETAILS

NUMBER OF VISITORS

73

NUMBER OF ACCOUNTS

31

NUMBER OF CLICKS

120

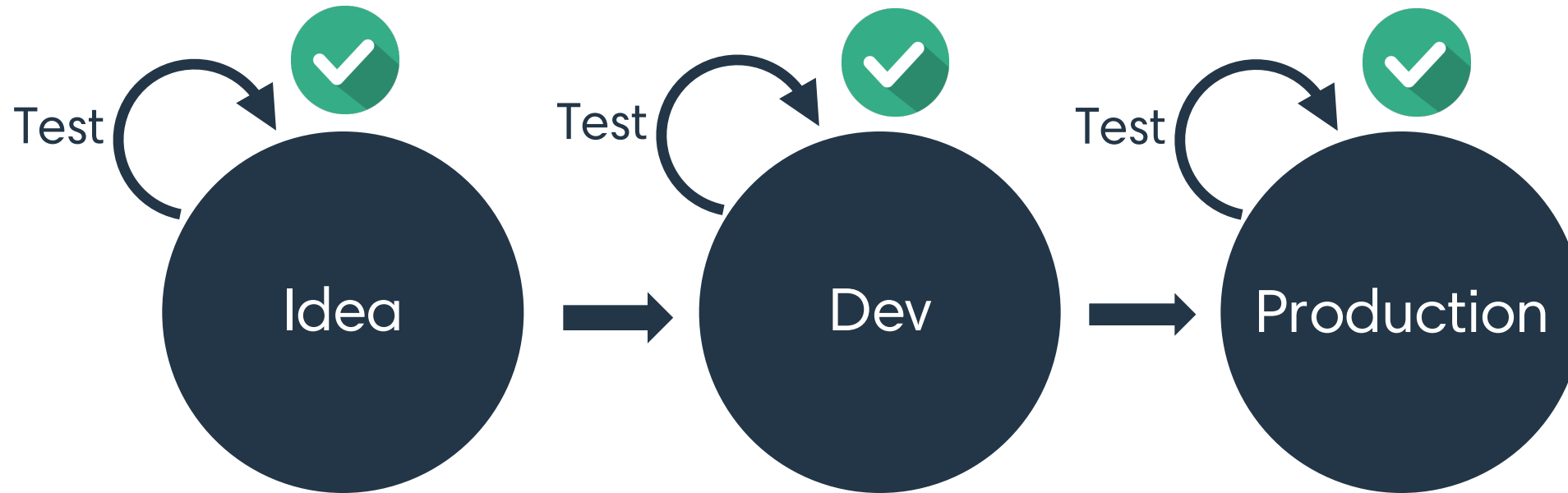
DAILY - FEATURE CLICKS -



A single place to collaborate on

- Feature behaviors
- Feature history
- Test results
- Feature usage & impact

Continuous testing



Test continuously your software from idea to production

Step 4 : Let's iterate

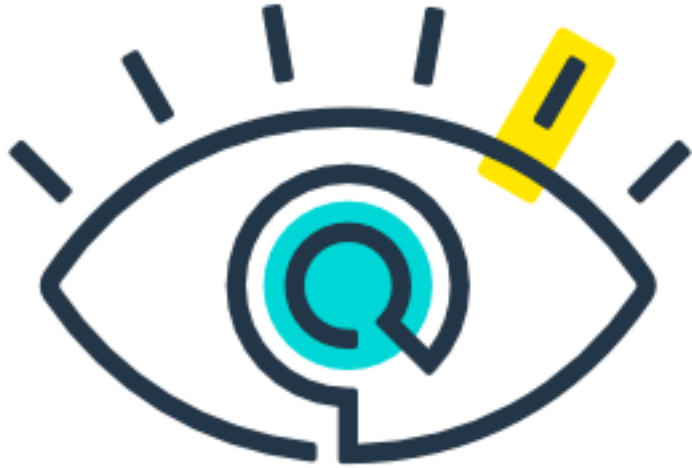
Let's iterate

1. In step 3 we have measured the impact
2. The feature can be removed (with the tests), enhanced or validated
3. Go to step 1



Conclusion

Benefits of testing in DevOps



1

Test value first

2

Incremental investment in quality

3

Context suitable to experiments

And remember...

**The biggest risk is
making something
that nobody want**



Thank you!



Laurent Py

@py_laurent
laurent@hiptest.com
<https://hiptest.com>



Vincent Pretre

vincent@hiptest.com
<https://hiptest.com>