

# 01 Introduction

Pourquoi, comment ?

Thibaut HENIN

[www.arsouyes.org](http://www.arsouyes.org)

# Ariane 5

<https://www.youtube.com/watch?v=fCn0-UYF3co>



# Ariane 5

Divulgâchage : pourquoi tester ?

# Ariane 5

Pourquoi ce comportement ?

# Ariane 5

Pourquoi ce comportement ?

## Explosion

(autodestruction)

# Ariane 5

Pourquoi ce comportement ?

**Explosion**

(autodestruction)

**Un booster s'est détaché**

(la fusée est en dérapage  $> 20^\circ$ )

# Ariane 5

Pourquoi ce comportement ?

**Explosion**

(autodestruction)

**Un booster s'est détaché**

(la fusée est en dérapage  $> 20^\circ$ )

**Le pilote automatique a ordonné un virage**

(données inertielles incohérentes)

# Ariane 5

Pourquoi ces données ?



# Ariane 5

Pourquoi ces données ?

## Centrales InertielleS HS

(SRI2 transmet un diagnostique)

# Ariane 5

Pourquoi ces données ?

## Centrales InertielleS HS

(SRI2 transmet un diagnostique)

## Operand Error : float64 vers int16

(Biais Horizontal plus grand qu'attendu)

# Ariane 5

Pourquoi ces données ?

```
P_M_DERIVE(T_ALG.E_BH) :=  
    UC_16S_EN_16NS(  
        TDB.T_ENTIER_16S(  
            (1.0 / C_M_LSB_BH) *  
            G_M_INFO_DERIVE(T_ALG.E_BH)  
        )  
    )
```

# Ariane 5

Pourquoi ces données ?

```
P_M_DERIVE(T_ALG.E_BH) :=  
    UC_16S_EN_16NS(  
        TDB.T_ENTIER_16S(  
            (1.0 / C_M_LSB_BH) *  
            G_M_INFO_DERIVE(T_ALG.E_BH)  
        )  
    )
```

# Ariane 5

Pourquoi ces données ?

```
P_M_DERIVE(T_ALG.E_BH) :=  
    UC_16S_EN_16NS(  
        TDB.T_ENTIER_16S(  
            (1.0 / C_M_LSB_BH) *  
            G_M_INFO_DERIVE(T_ALG.E_BH)  
        )  
    )
```

# Ariane 5

Pourquoi ces données ?

```
P_M_DERIVE(T_ALG.E_BH) :=  
    UC_16S_EN_16NS(  
        TDB.T_ENTIER_16S(  
            (1.0 / C_M_LSB_BH) *  
            G_M_INFO_DERIVE(T_ALG.E_BH)  
        )  
    )
```

# Ariane 5 - patch

Version corrigée

```
L_M_BH_32 := TDB.T_ENTIER_32S(  
    (1.0 / C_M_LSB_BH) *  
    G_M_INFO_DERIVE(T_ALG.E_BH)  
)  
  
if L_M_BH_32 > 32767 then  
    P_M_DERIVE(T_ALG.E_BH) := 16#7FFF# ;  
elseif L_M_BH_32 < -32768 then  
    P_M_DERIVE(T_ALG.E_BH) := 16#8000# ;  
else  
    P_M_DERIVE(T_ALG.E_BH) := UC_16S_EN_16NS(  
        TDB.T_ENTIER_16S(L_M_BH_32)  
    ) ;  
end if;
```

# Ariane 5

Pourquoi ce fonctionnement ?



# Ariane 5

Pourquoi ce fonctionnement ?

## Reprise d'Ariane 4

(matériel, logiciel et procédure)

# Ariane 5

Pourquoi ce fonctionnement ?

**Reprise d'Ariane 4**

(matériel, logiciel et procédure)

**Pas (re) testé**

(économie de 800 000 Fr)

# Mais bien d'autres...

<https://www.rocketprojet.com/29-bugs-informatiques-catastrophe/>

2009, Toyota, 200 à 400 morts et 2,3 G€

(accélérateur a fond, frein désactivé)

2001, Institut National du Cancer, 17 morts, 11 blessés

(calcul de la dose de radiation erronés)

2012, Knight Capital Group, 410 M€

(Trading inverse achat et vente)

# Pour qui ?

Tout est question de point de vue

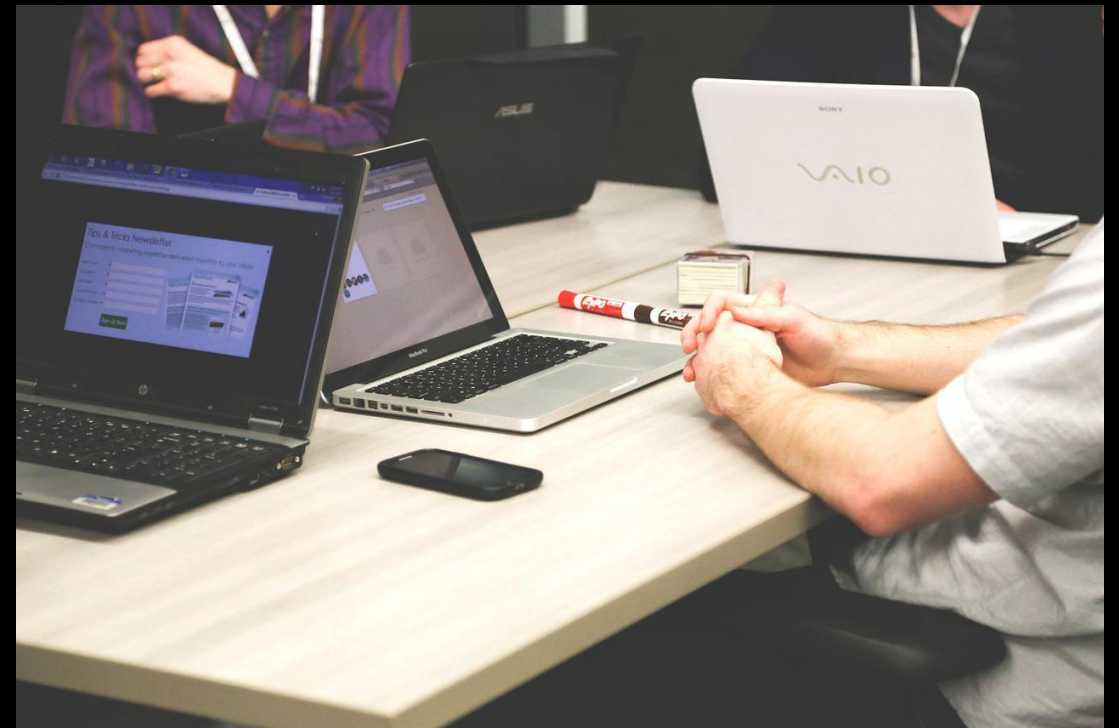
# Client vs prestataire

Recette & Maintenance

(prévue au contrat)

Sinon délais légaux

(paiement & garanties)



© StartupStockPhotos @ pixabay

# Etat vs industriels

## Normes

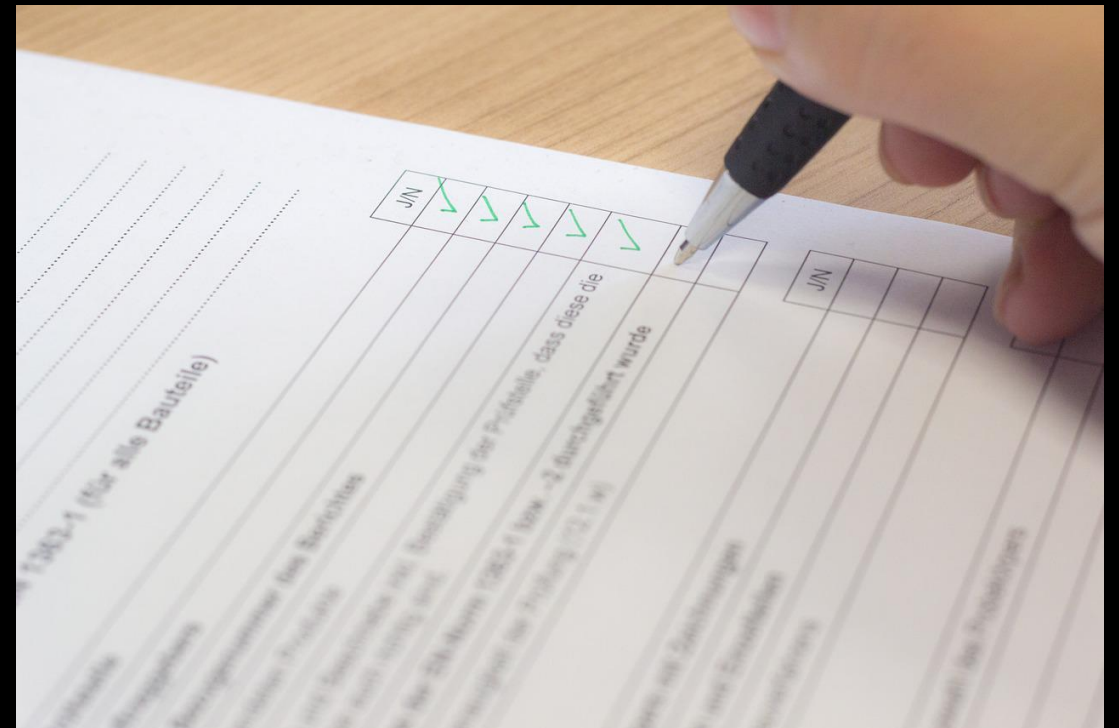
(DO-178C, PCI-DSS, CC, ...)



© moritz320 @ pixabay

# Editeur vs développeurs

Département « Q&A »  
(pour l'image de marque)



© DCG\_MAK @ pixabay



# Développeurs vs éditeurs

« Bonnes pratiques »  
(efficacité & productivité)



© PublicDomainPictures @ pixabay



# Synthèse

	Exigé par le Client	Exigé par L'Editeur
Exigences Spécifique	Recettes	Q&A
Exigences Générale	Normes	Bonnes pratiques

# Comment tester ?

Pour qui tester ?

# Développer un logiciel

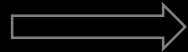
## Processus

Besoin

# Développer un logiciel

## Processus

Besoin



Développement

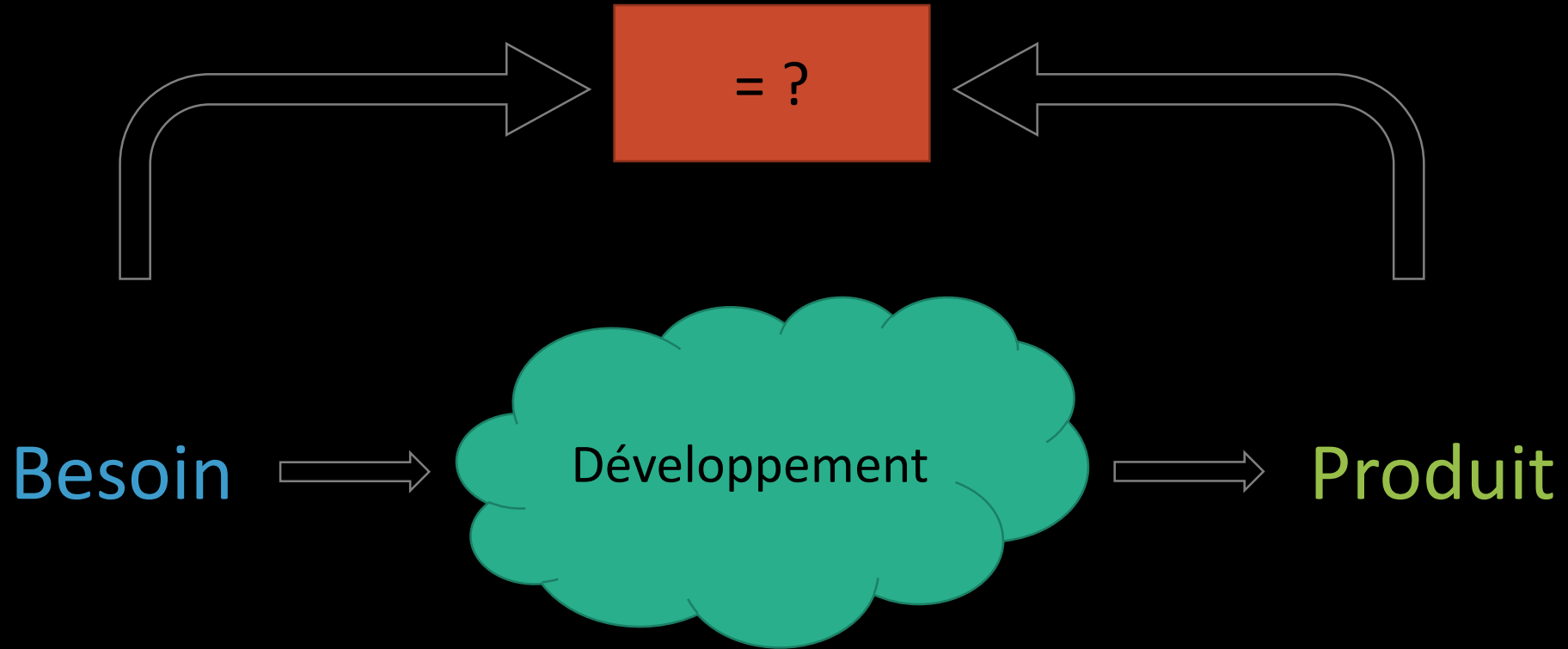
# Développer un logiciel

## Processus



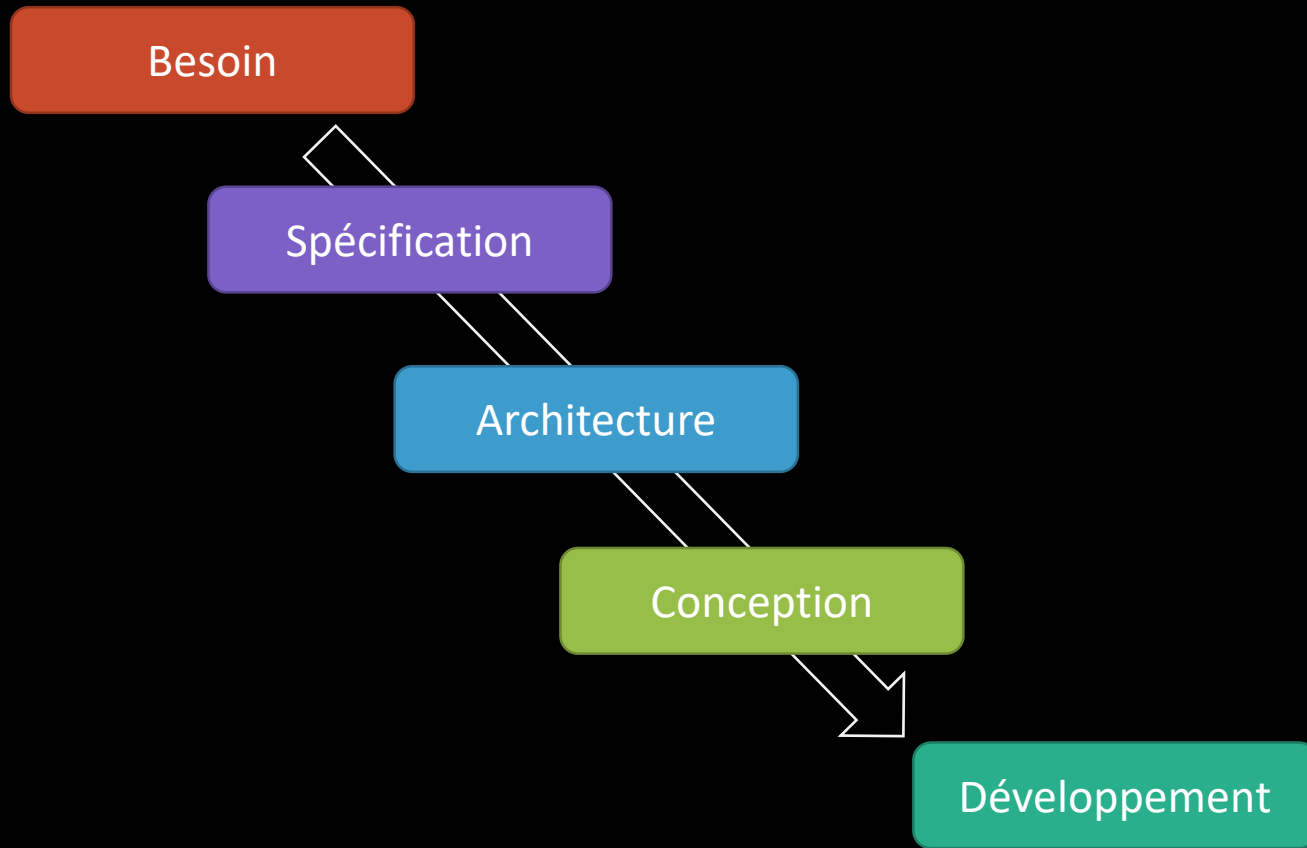
# Tester un logiciel

## Processus



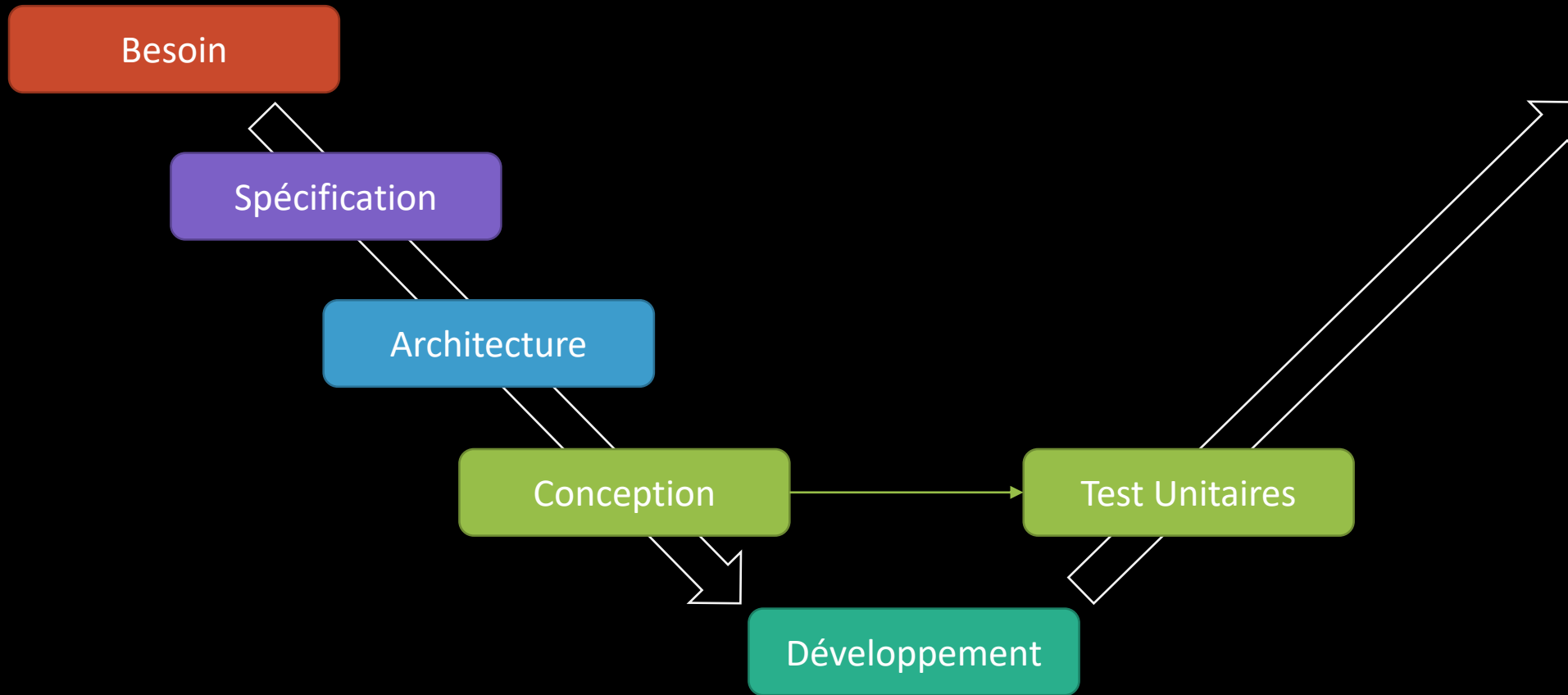
# Tester par jalons

## Cycle en V



# Tester par jalons

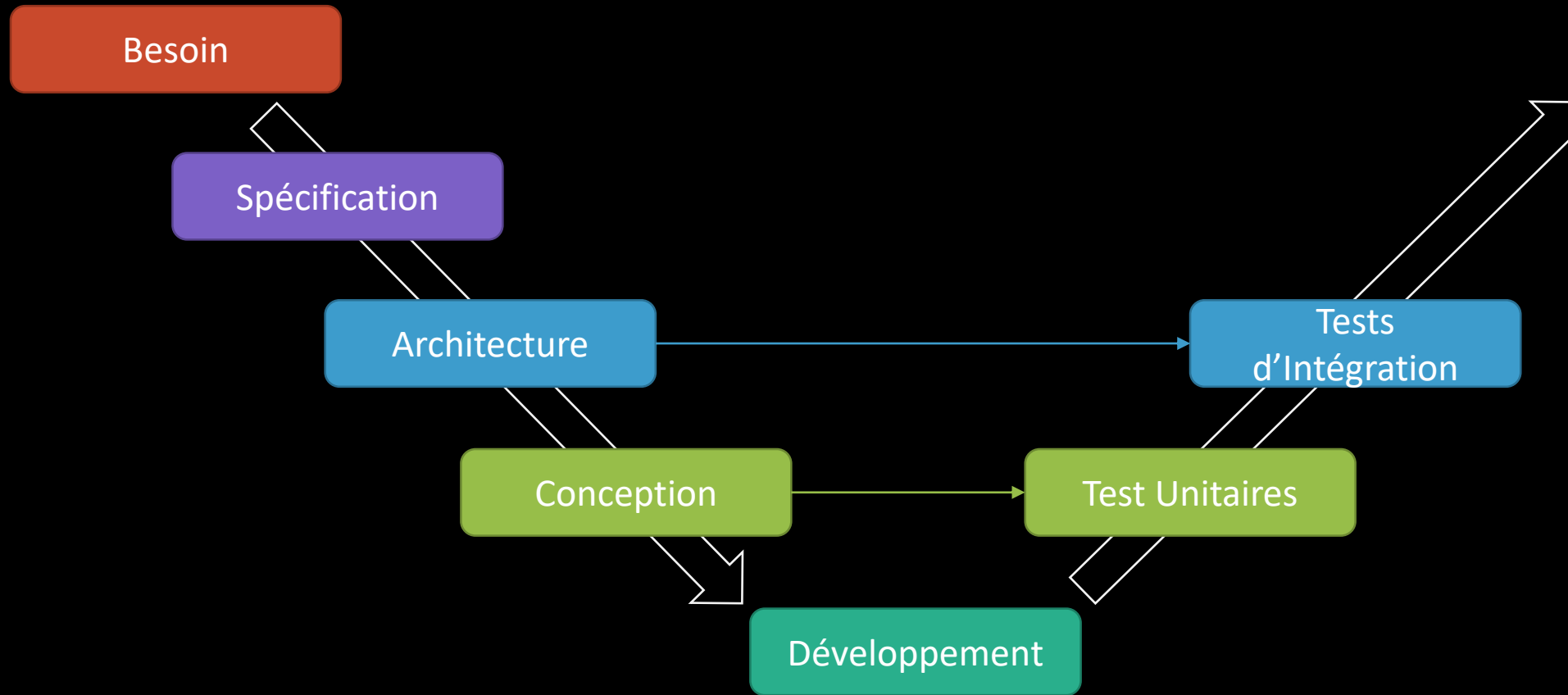
## Cycle en V





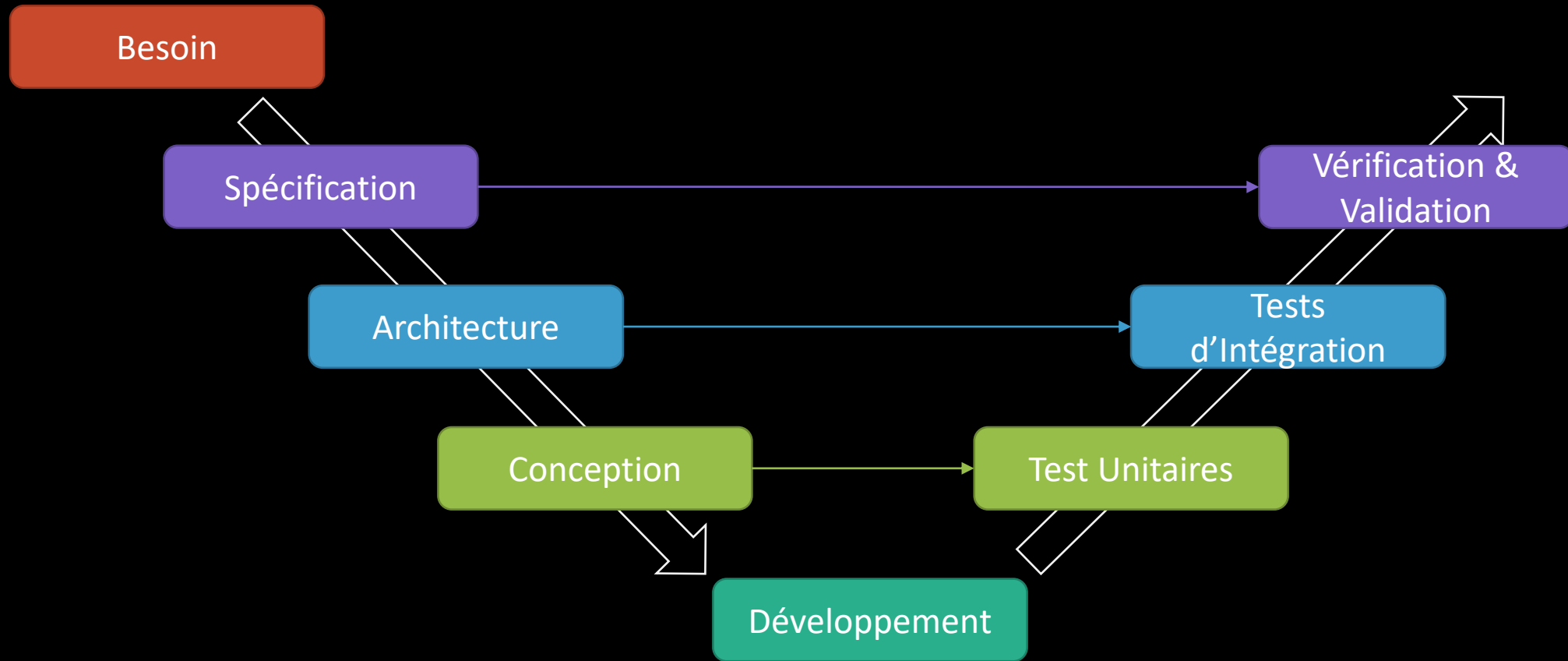
# Tester par jalons

## Cycle en V



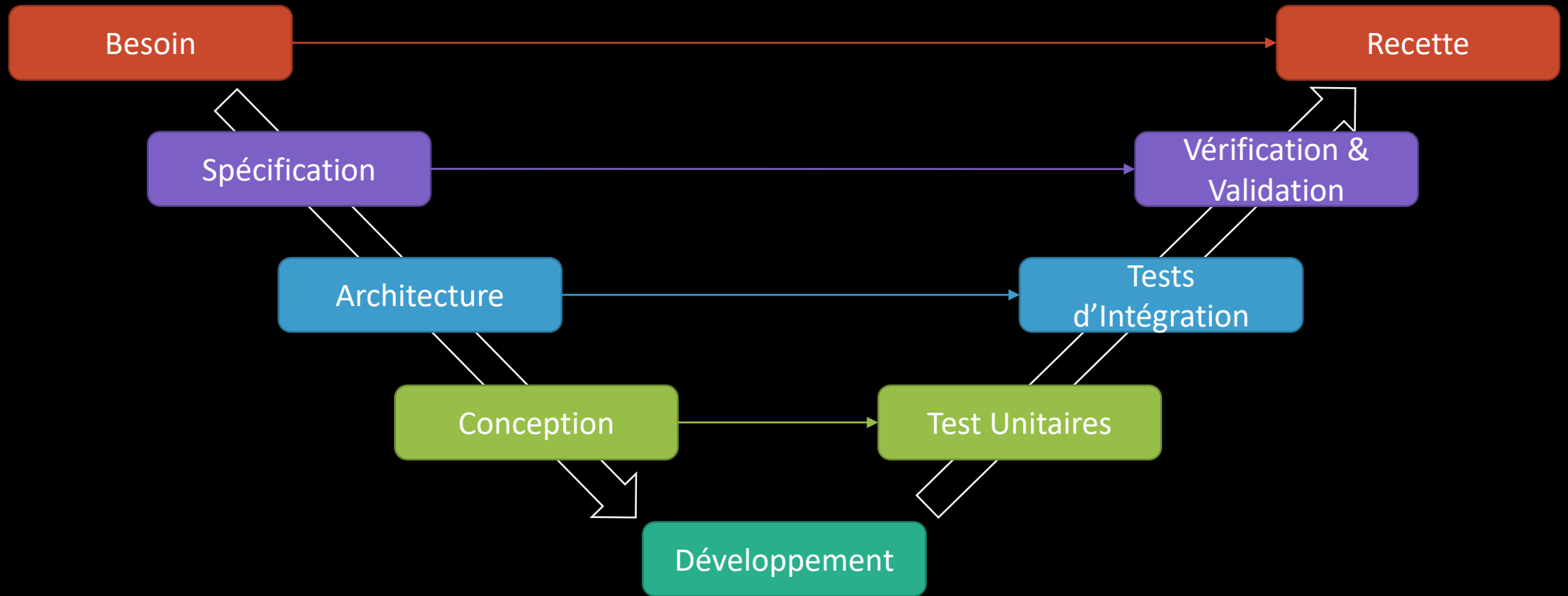
# Tester par jalons

## Cycle en V



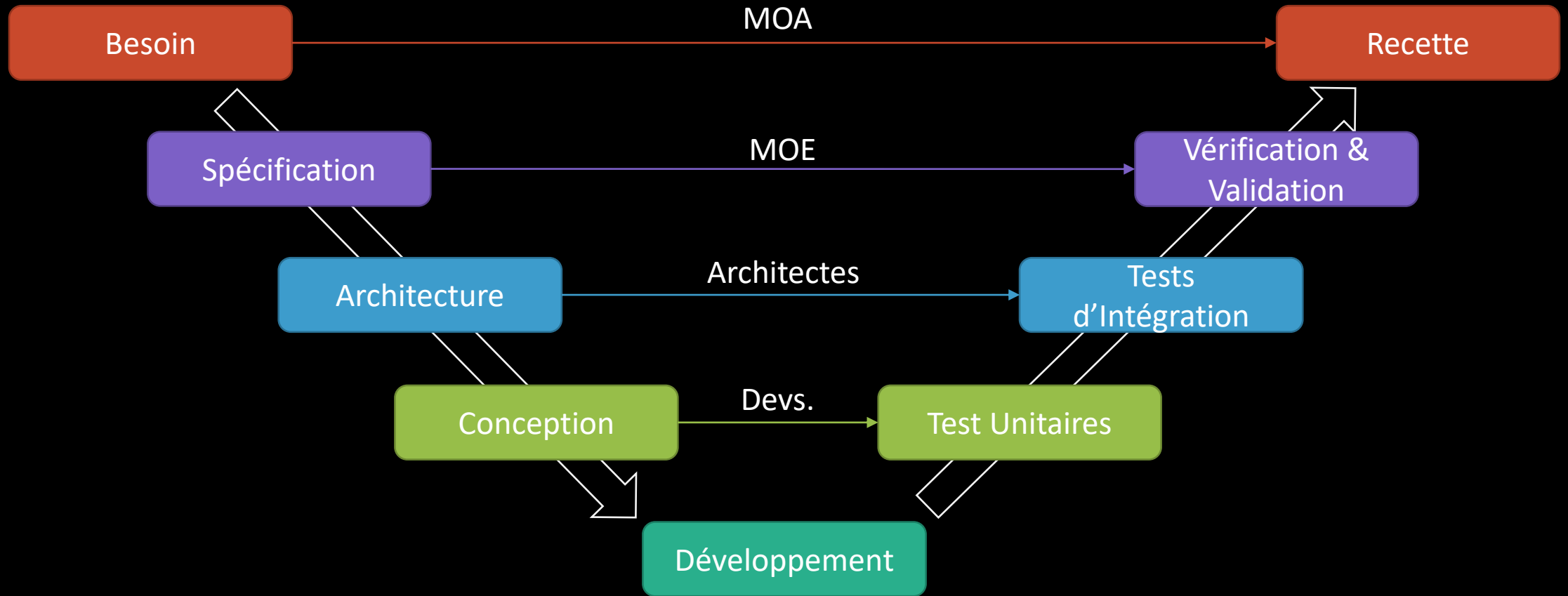
# Tester par jalons

## Cycle en V



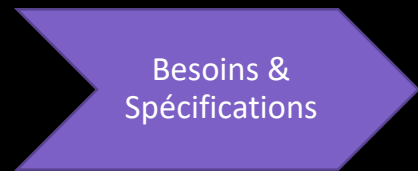
# Tester par jalons

## Cycle en V



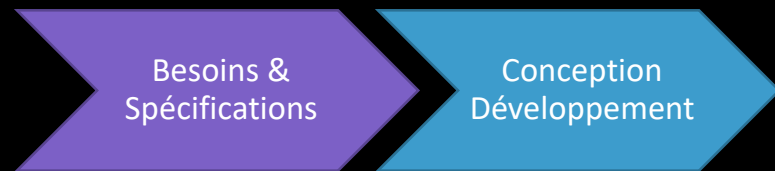
# Agilité & DevOps

Tests en continu, CI/CD



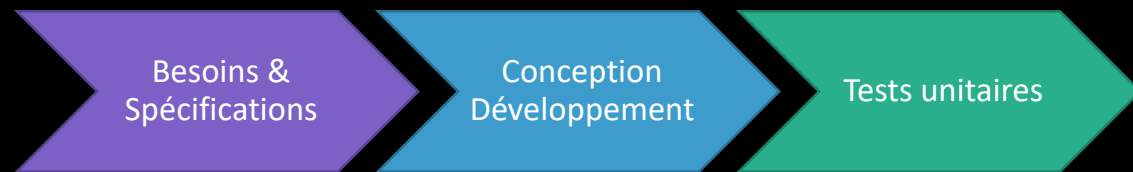
# Agilité & DevOps

Tests en continu, CI/CD



# Agilité & DevOps

Tests en continu, CI/CD



# Agilité & DevOps

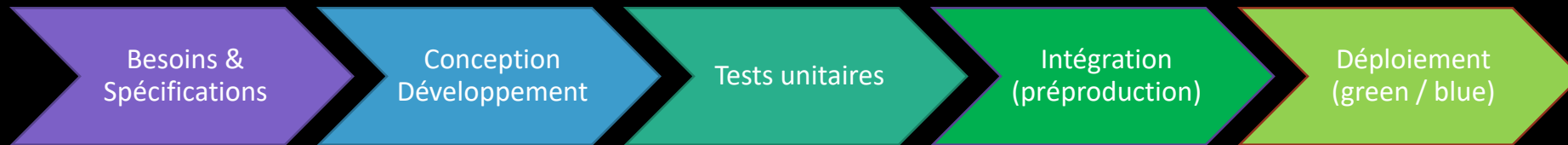
Tests en continu, CI/CD





# Agilité & DevOps

Tests en continu, CI/CD



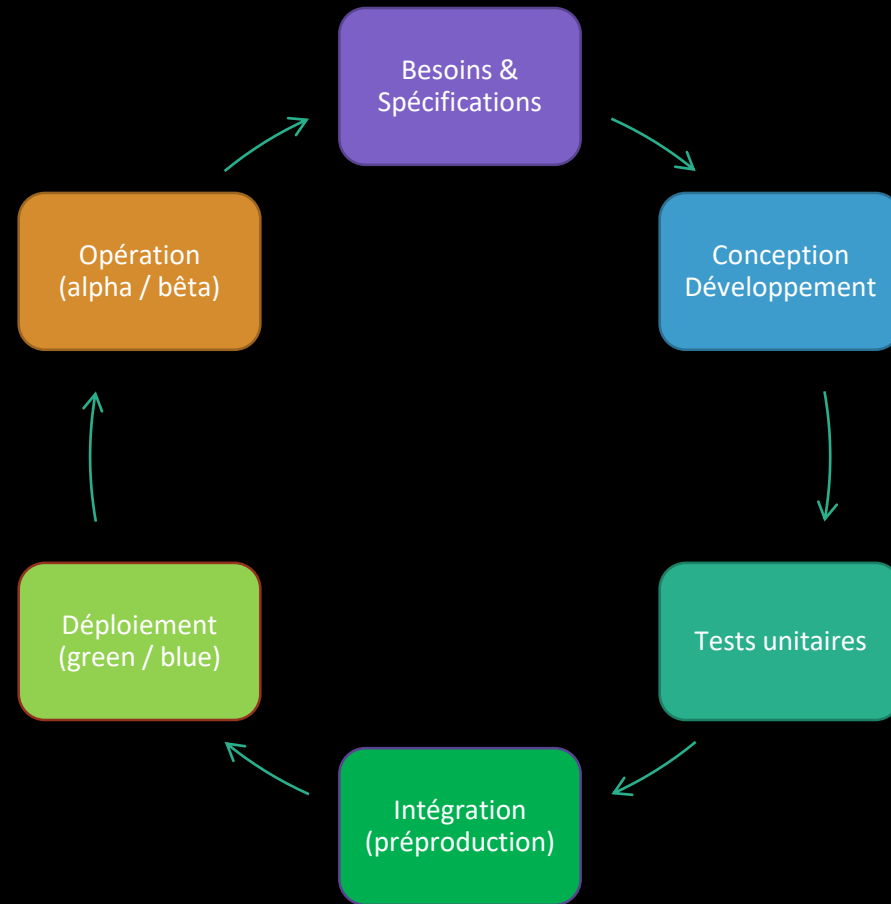
# Agilité & DevOps

## Tests en continu, CI/CD



# Agilité & DevOps

## Tests en continu, CI/CD



# Conclusion

Que retenir ?

# C'est un métier

## Et des enseignements spécifiques

**Activité spécifique**

(Vérifier et Valider)

**Outils spécifiques**

(sélénium, gatling, ...)

**Qualifications spécifiques**

(rigueur, organisation)

# Ce cours

Objectifs et contenus

Cours → discussions

(philosophie, pourquoi du comment)

Travaux pratiques

(tests unitaires en PHP)

To test is  
to doubt

