

# 05 - Maintenance

Remboursement et refactoring

Thibaut HENIN

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

# The QA Team Testing the Bulletproof Vest

[https://www.youtube.com/watch?v=\\_HCjf\\_kBHP0](https://www.youtube.com/watch?v=_HCjf_kBHP0)



Developer

QA Engineer

# La Maintenance

Théorie de l'évolution

# Deux types

Corrective

Évolutive

# Maintenance corrective

*« Maintenance exécutée après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise »*

Norme NF EN 13306 X 60-319 de juin 2011

# Maintenance corrective

*Corriger les bogues*

# Maintenance évolutive

## Nouvelles fonctions

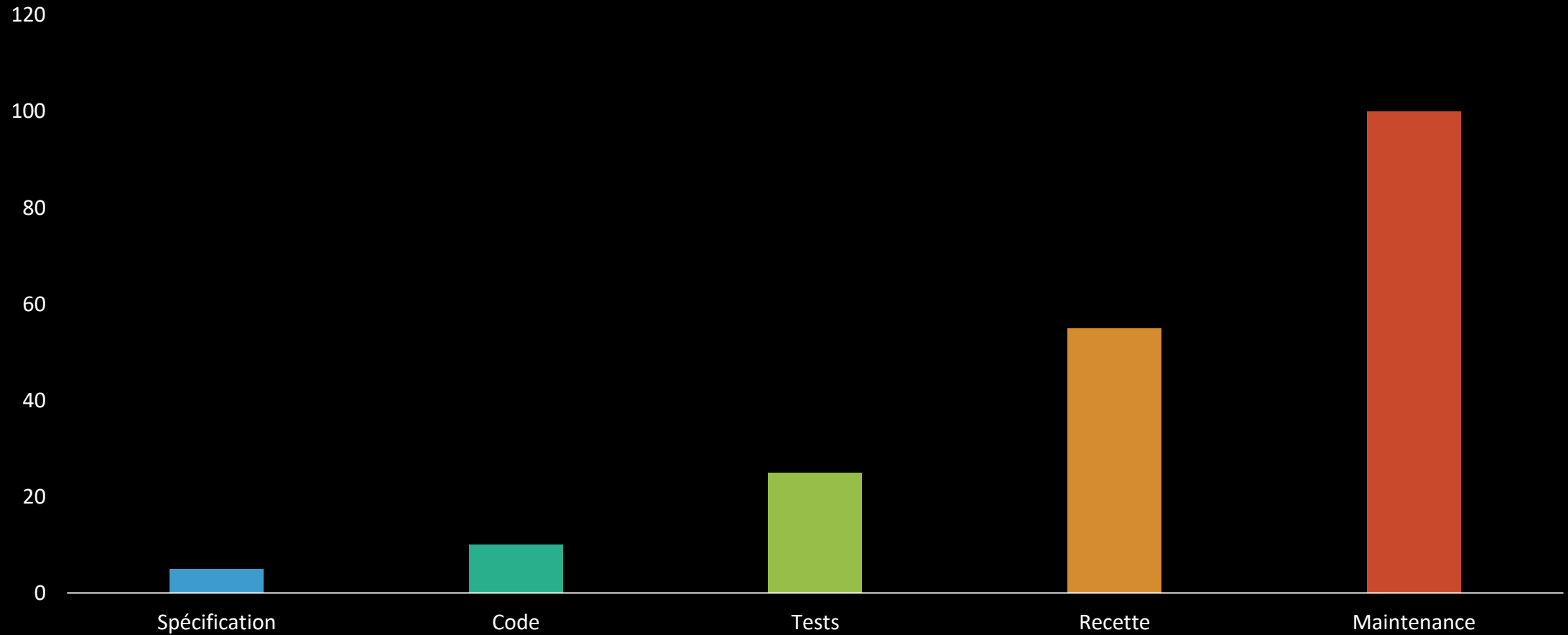
(demandes clients, évolution du marché)

## Adaptation de l'existant

(évolution des normes)

# Coût de correction

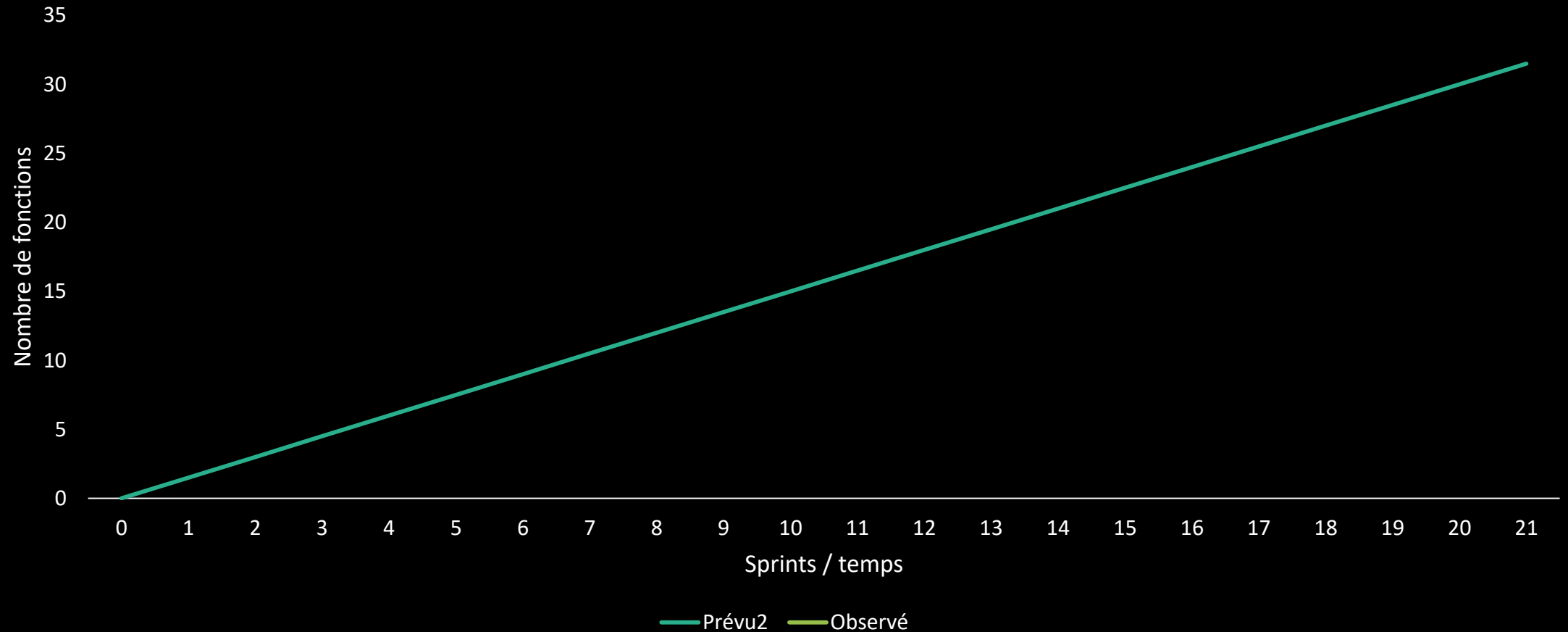
## Une légende urbaine ?!





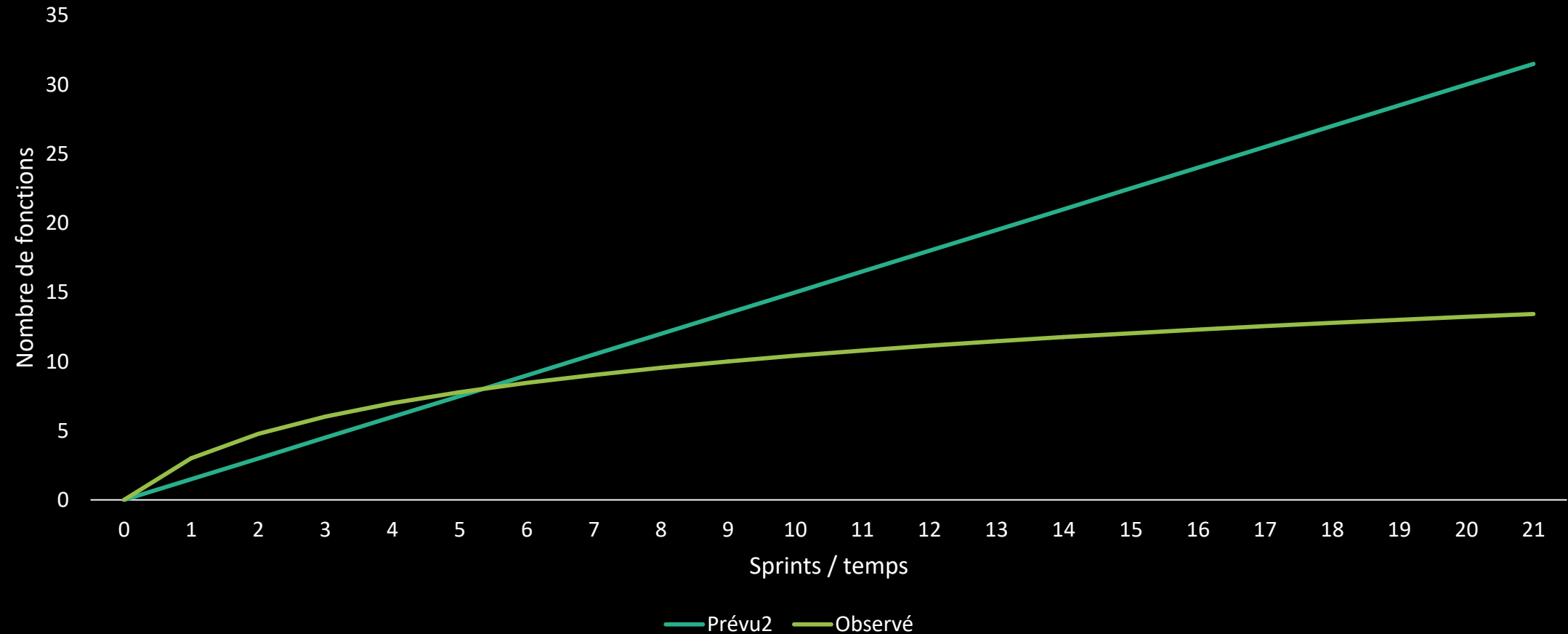
# Vitesse de développement

En théorie, là, ça devrait passer



# Vitesse de développement

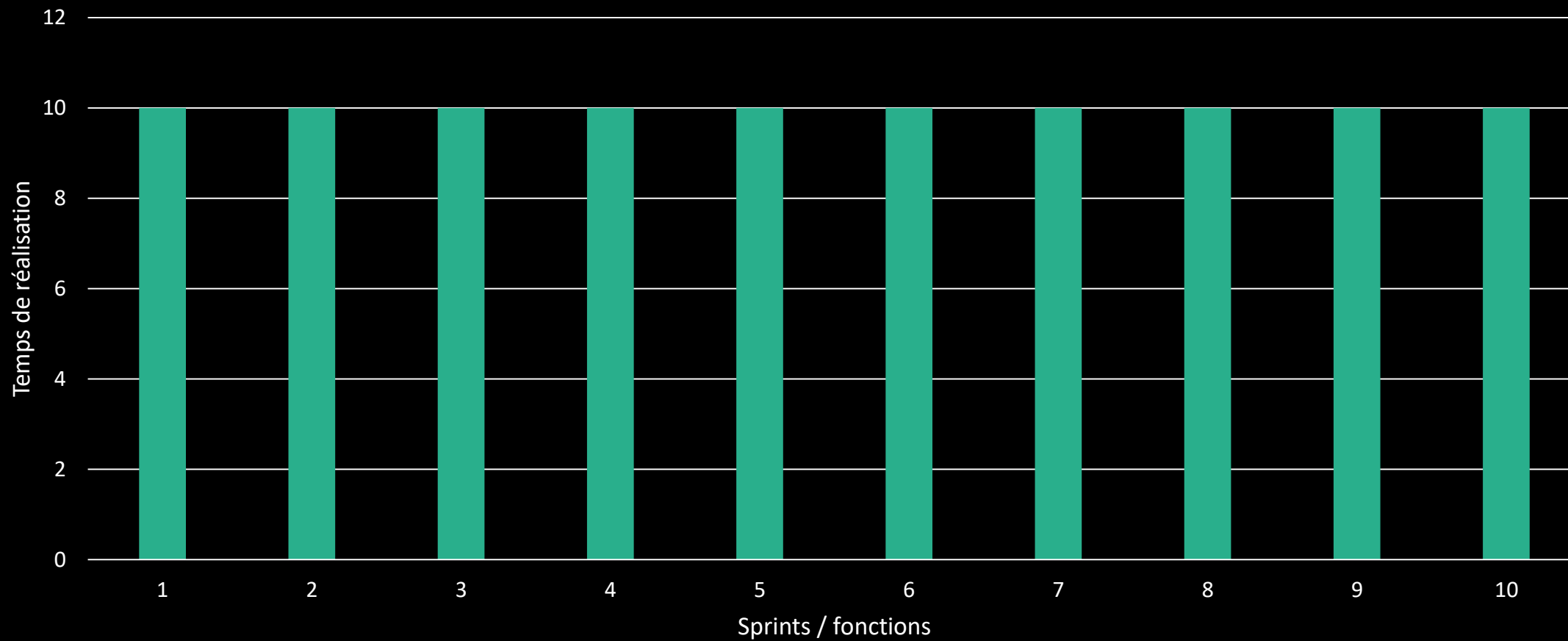
Plus ça va, moins ça va



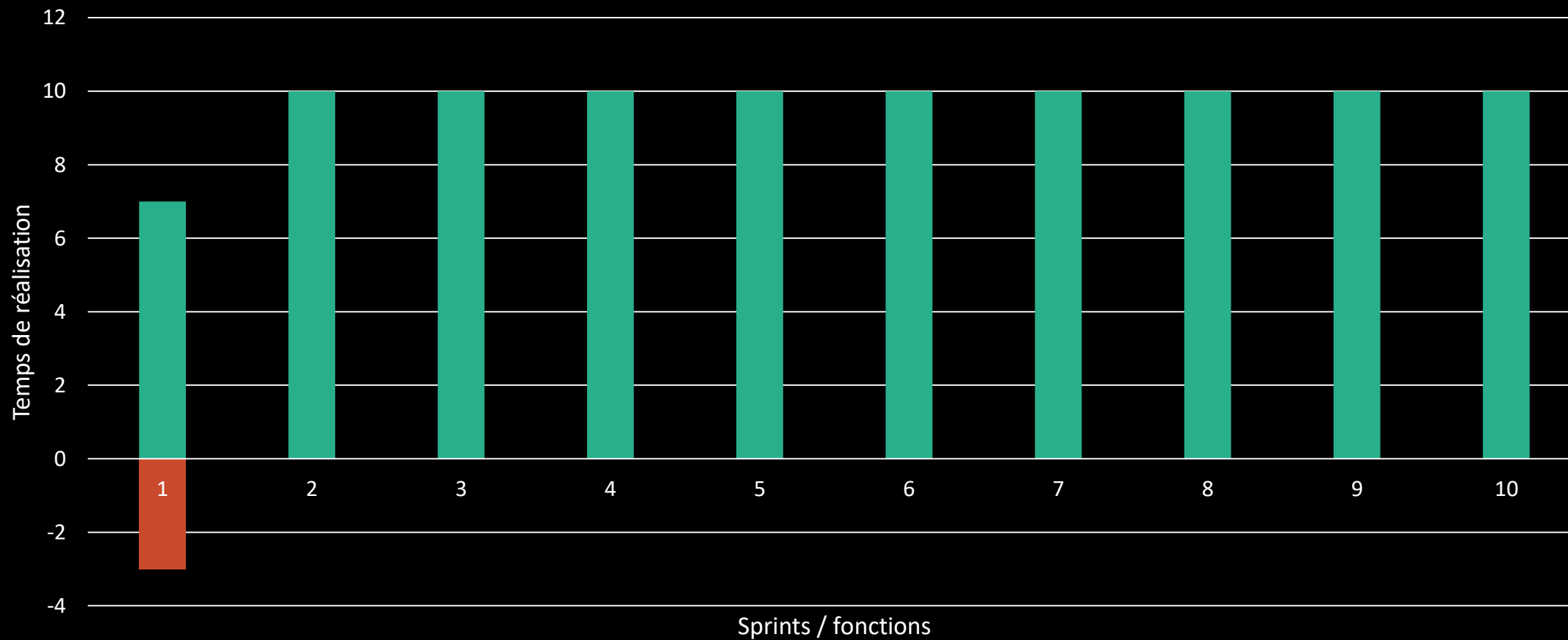
# Dette Technique

Intérêts et agios

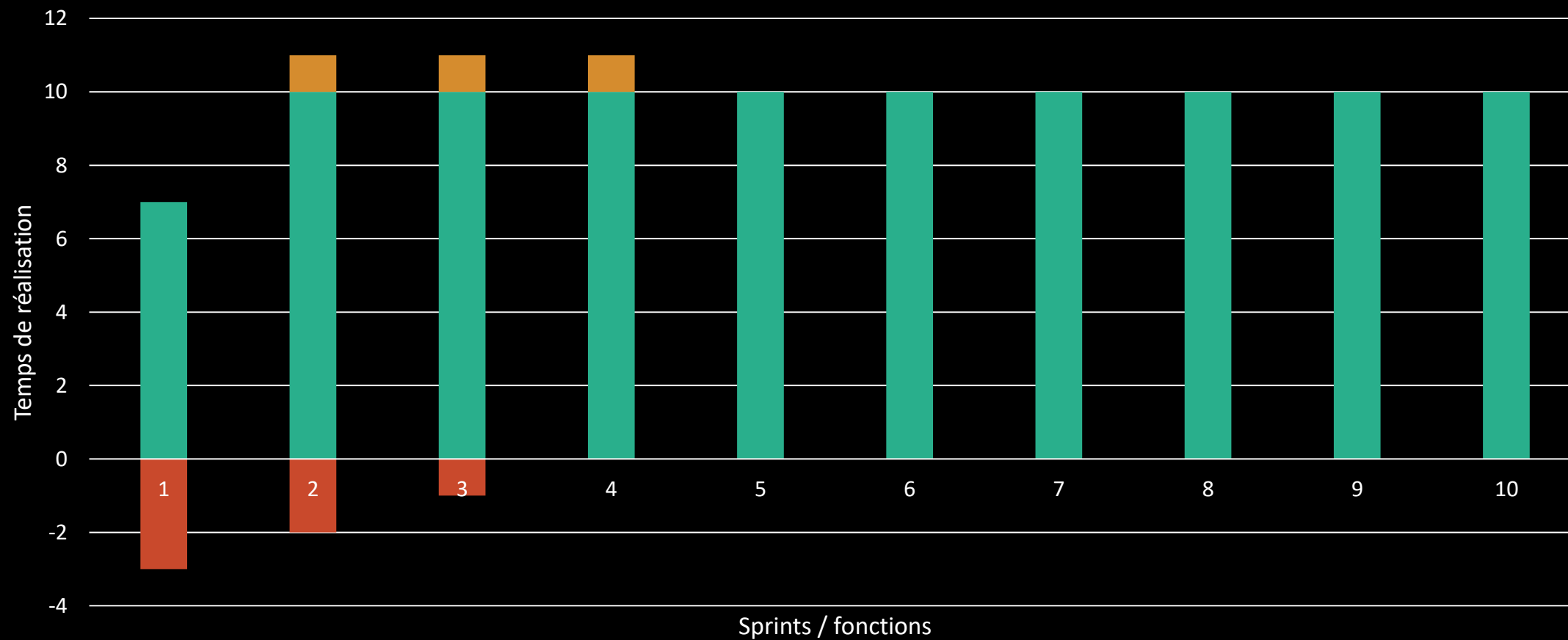
# Coût du travail bien fait



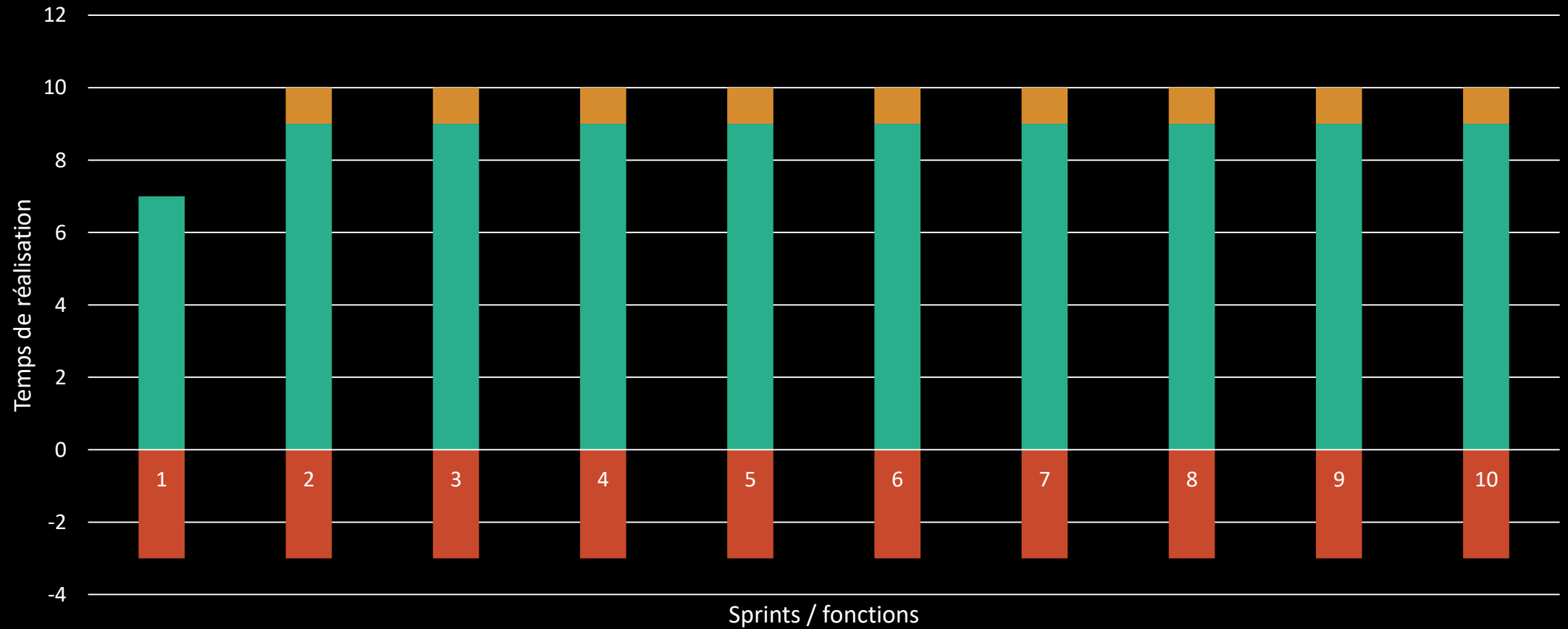
# Prêt d'accélération



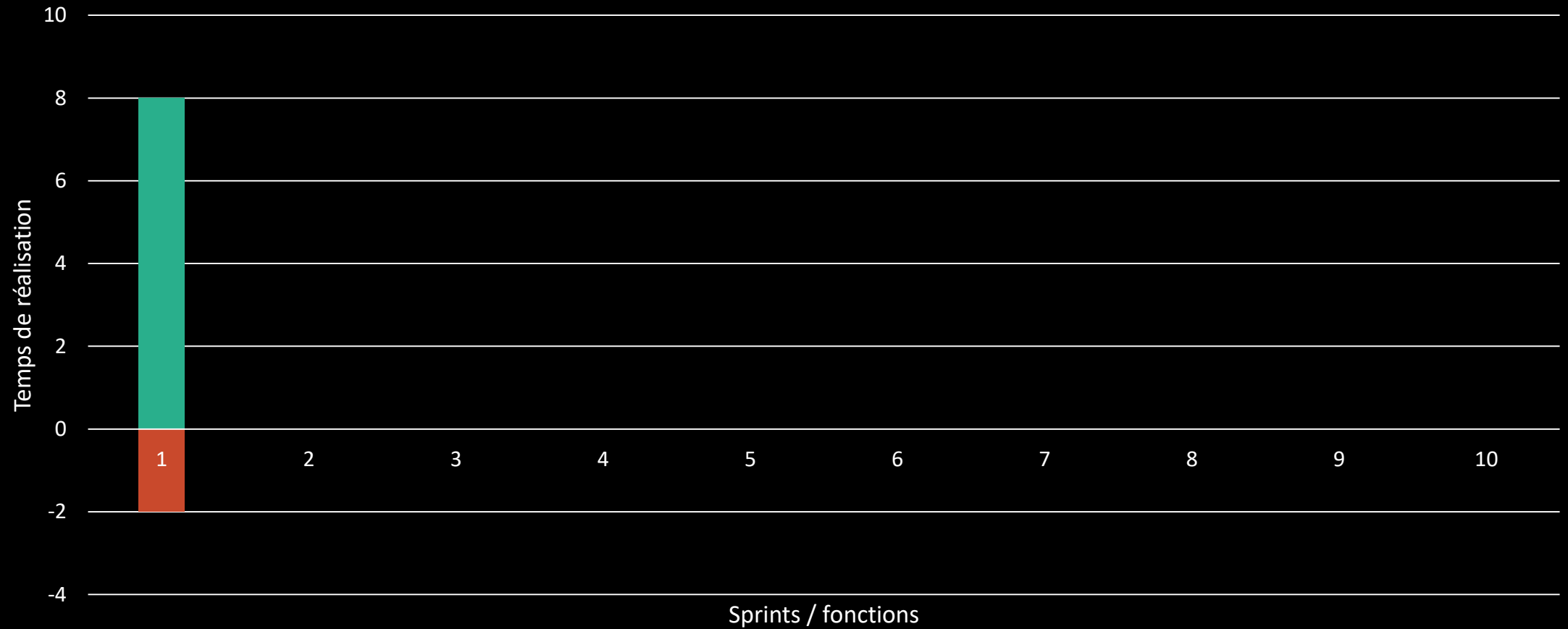
# Remboursement des intérêts



# Ressources limitées

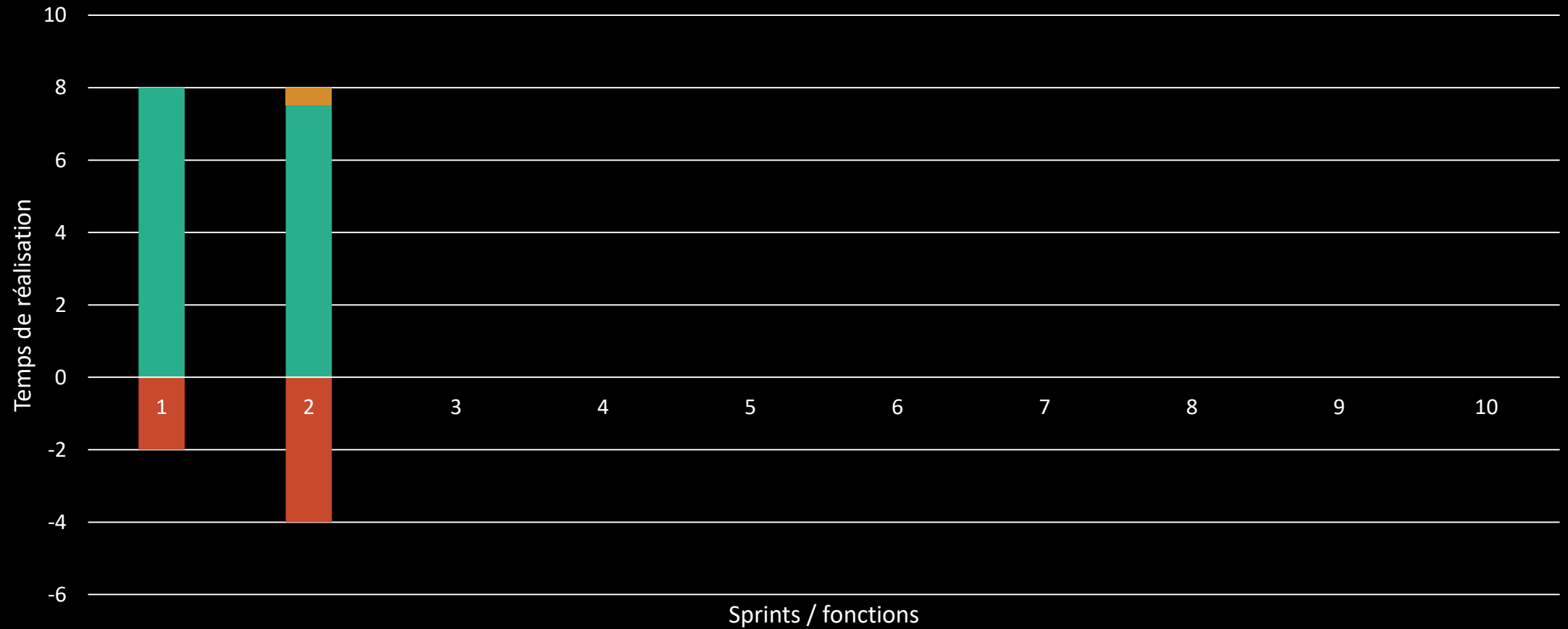


# Prendre de mauvaise habitude

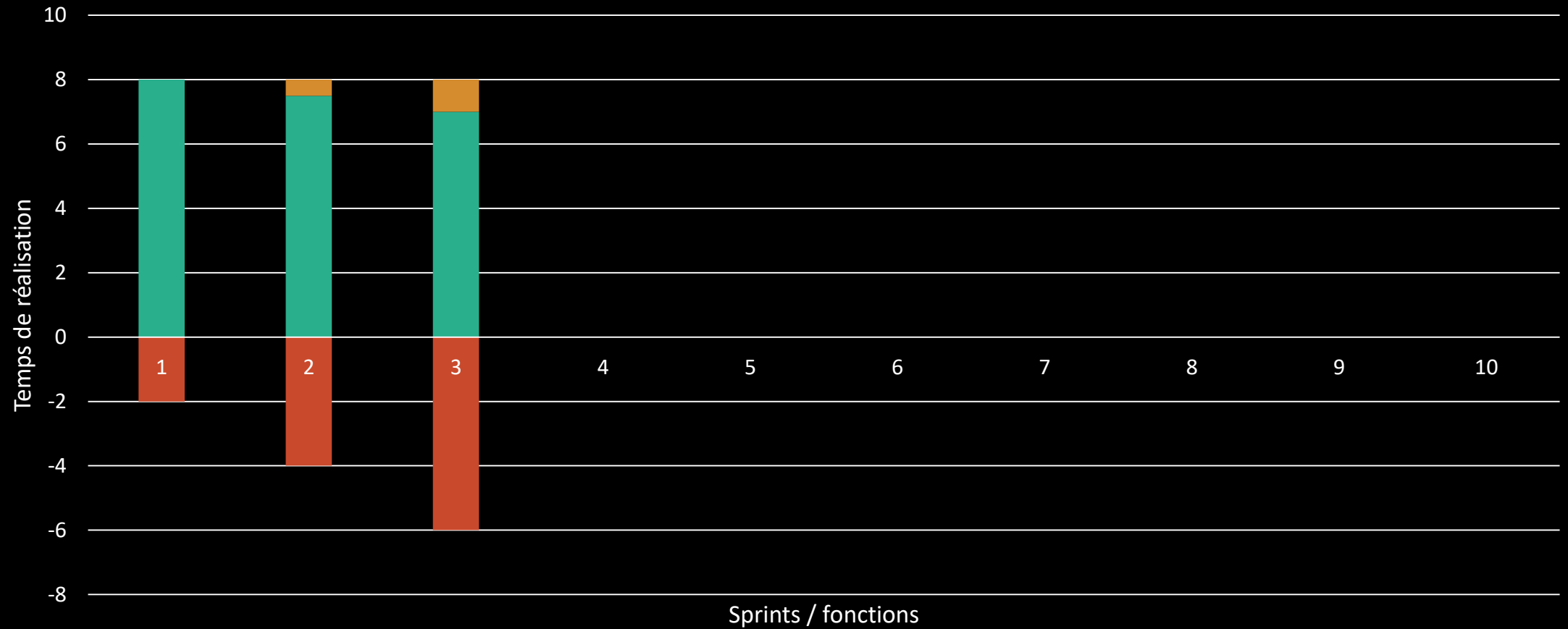




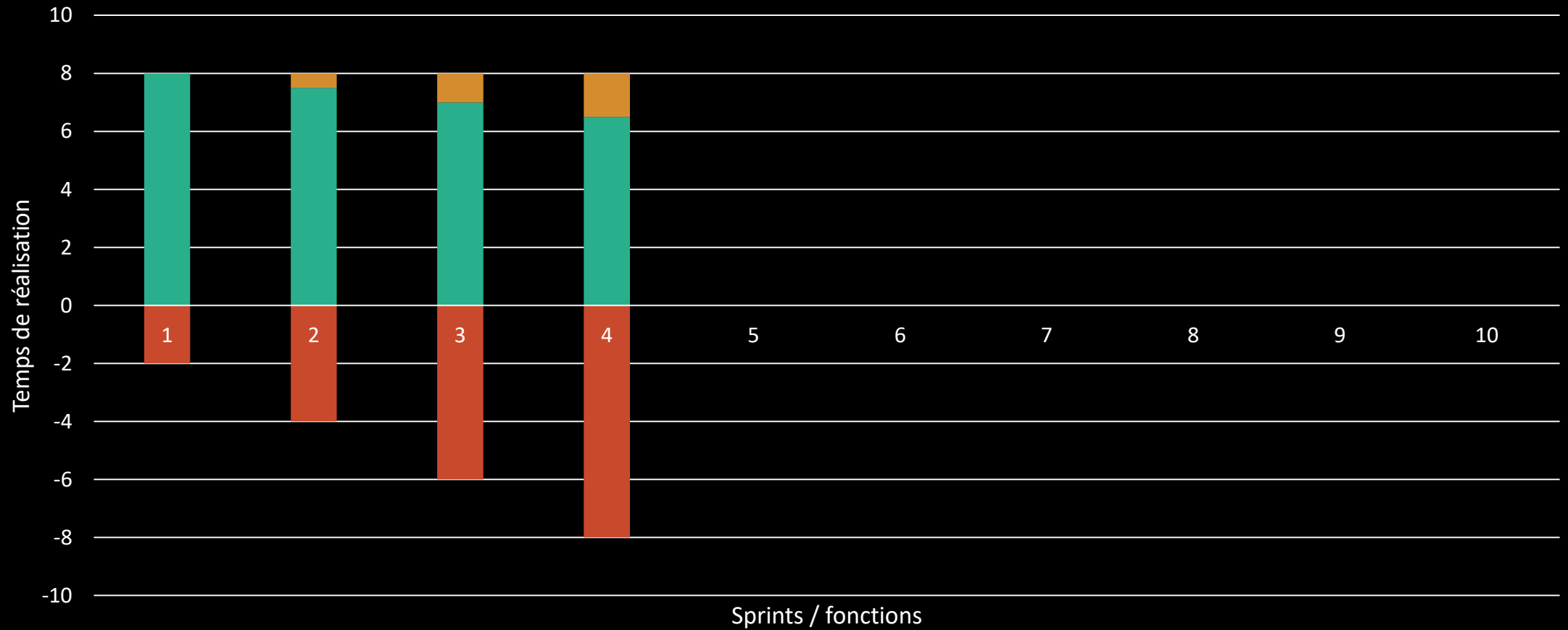
# Prendre de mauvaise habitude



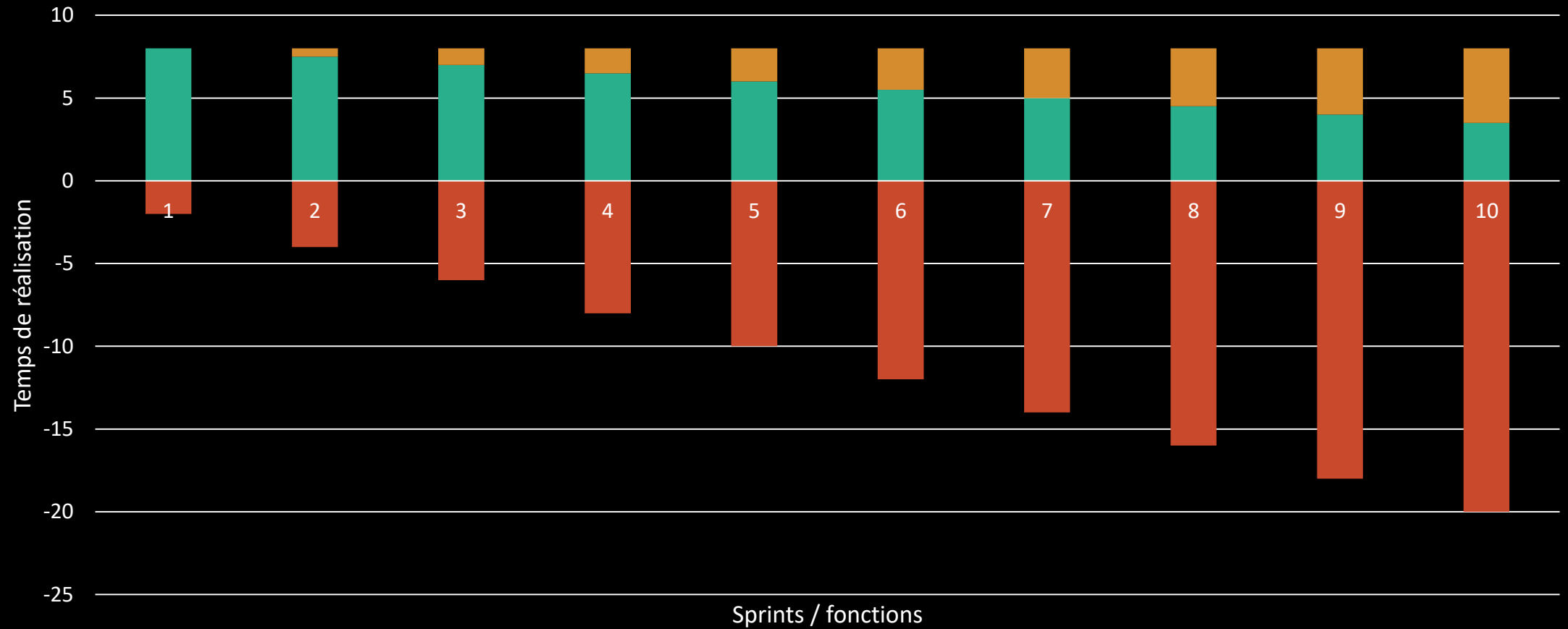
# Prendre de mauvaise habitude



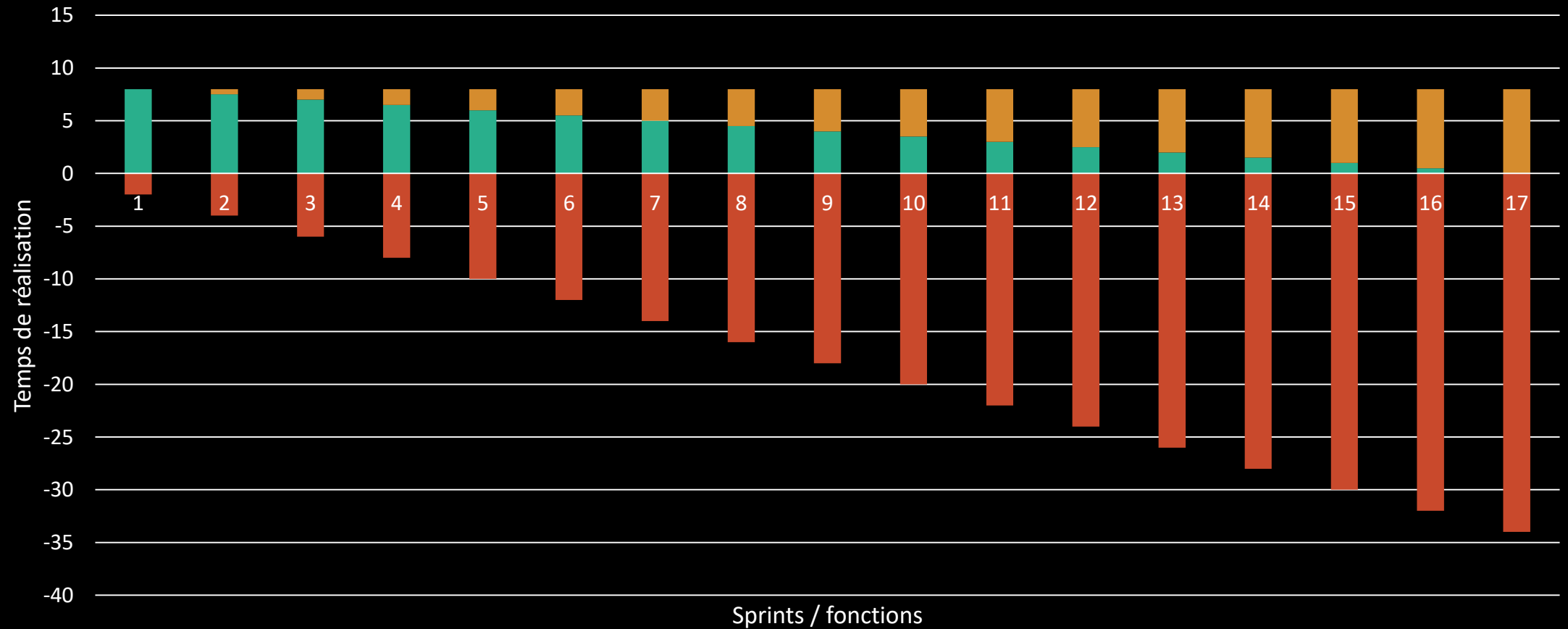
# Prendre de mauvaise habitude



# Prendre de mauvaise habitude

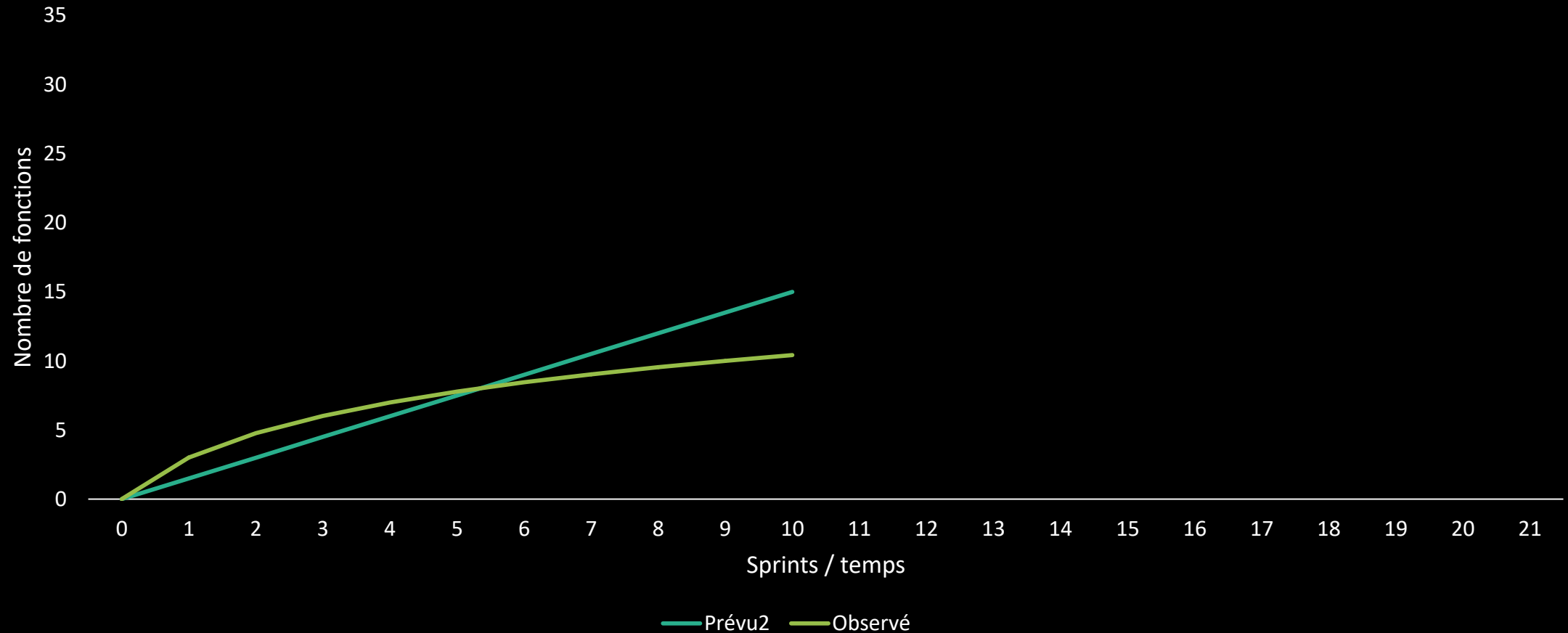


# Prendre de mauvaise habitude



# Choix après 10 sprints...

## Comment résoudre le problème ?

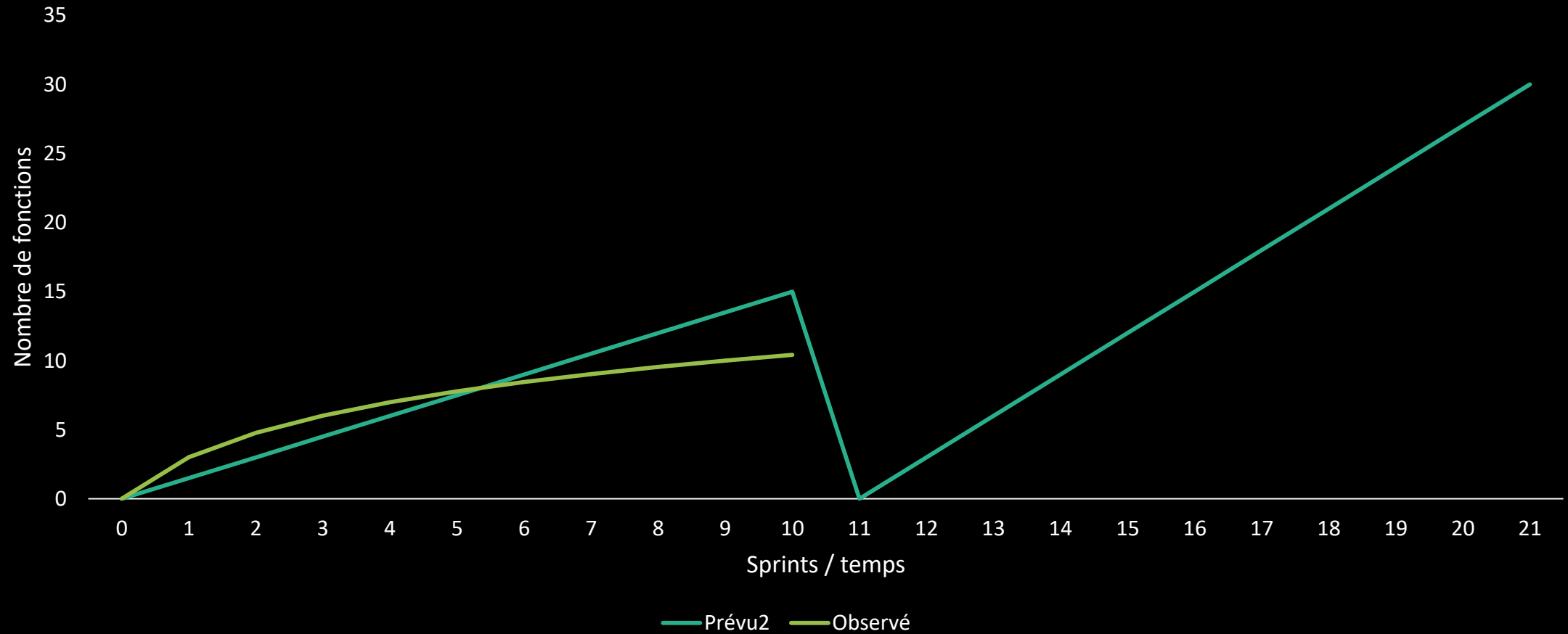


# Mauvaises solutions

Ça partait d'une bonne intention

# Reprendre à zéro

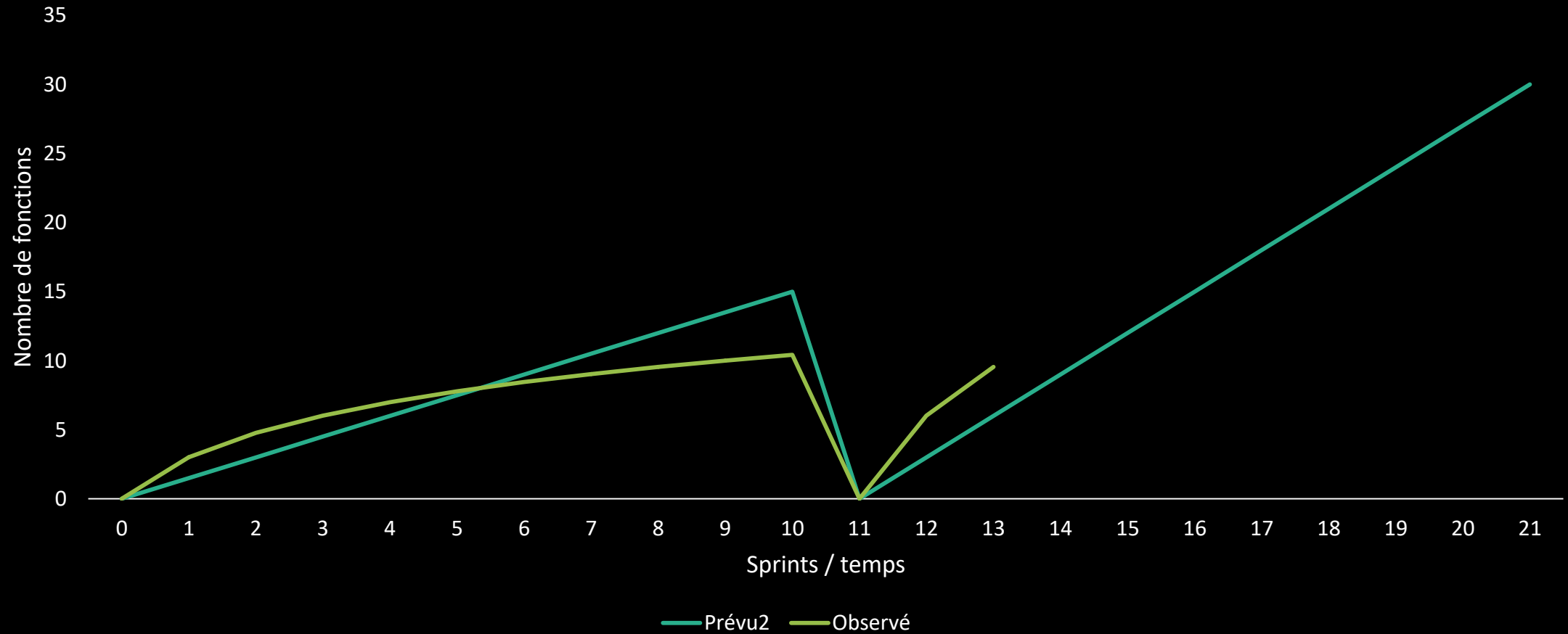
Quand ça va pas, ça va pas





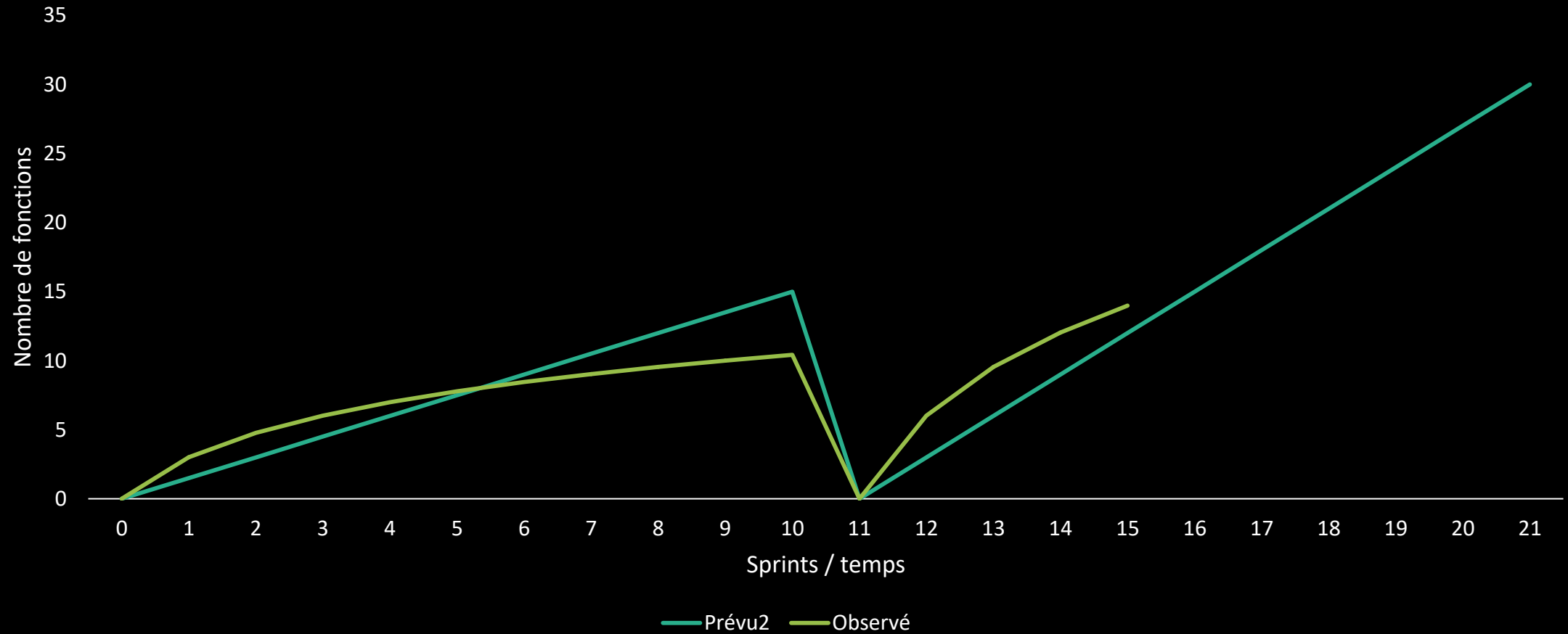
# Reprendre à zéro

Quand ça va pas, ça va pas



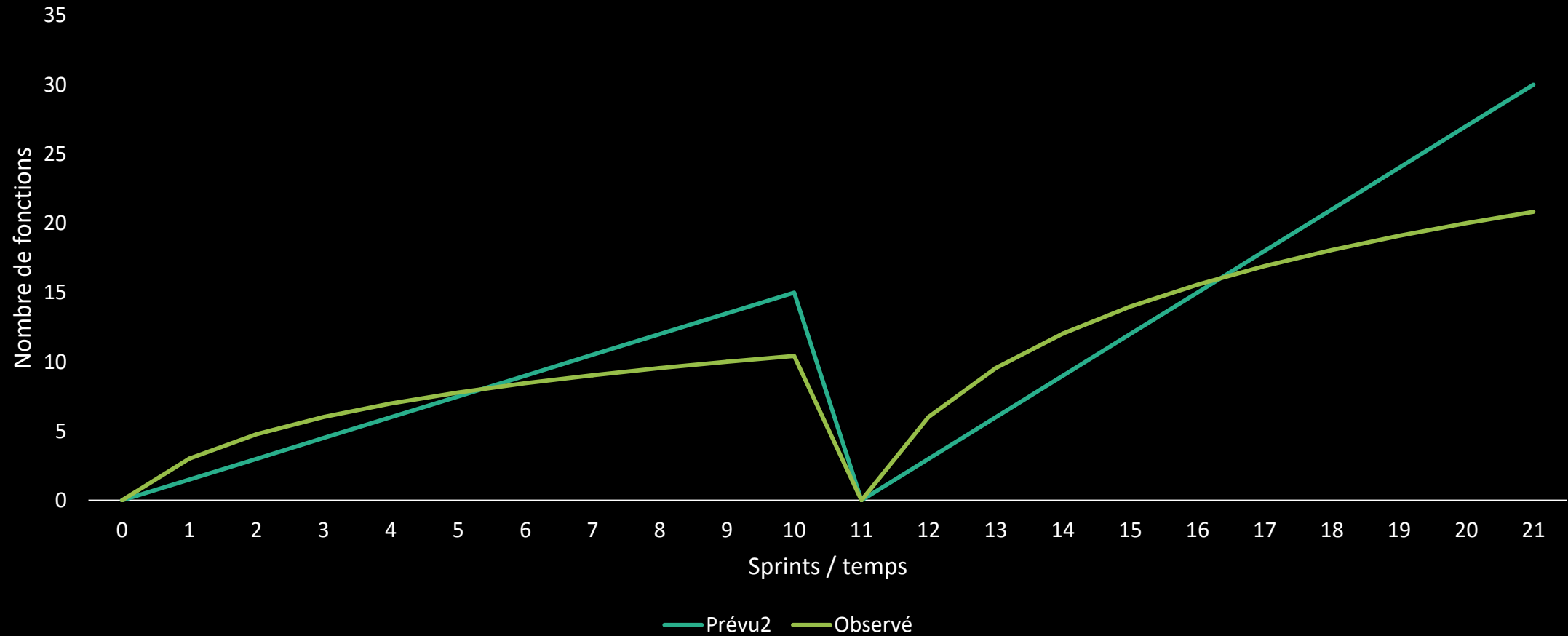
# Reprendre à zéro

Quand ça va pas, ça va pas



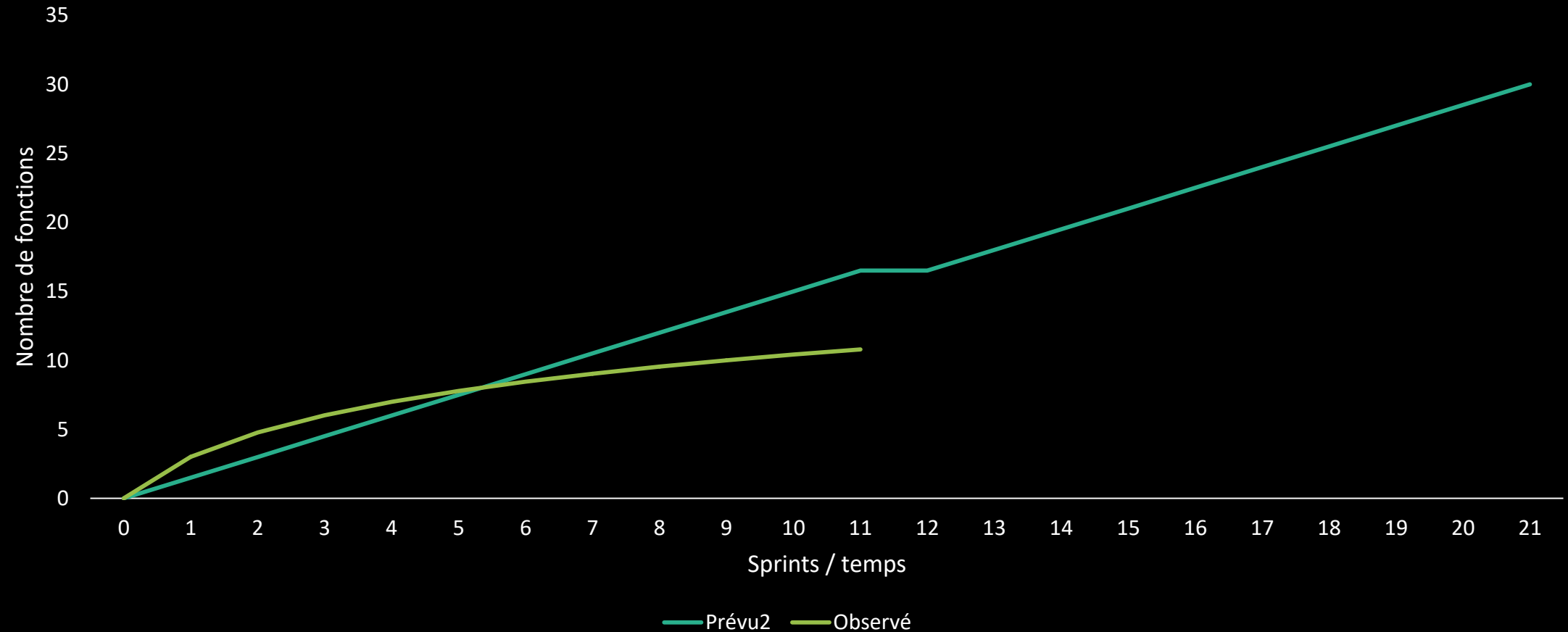
# Reprendre à zéro

Quand ça va pas, ça va pas



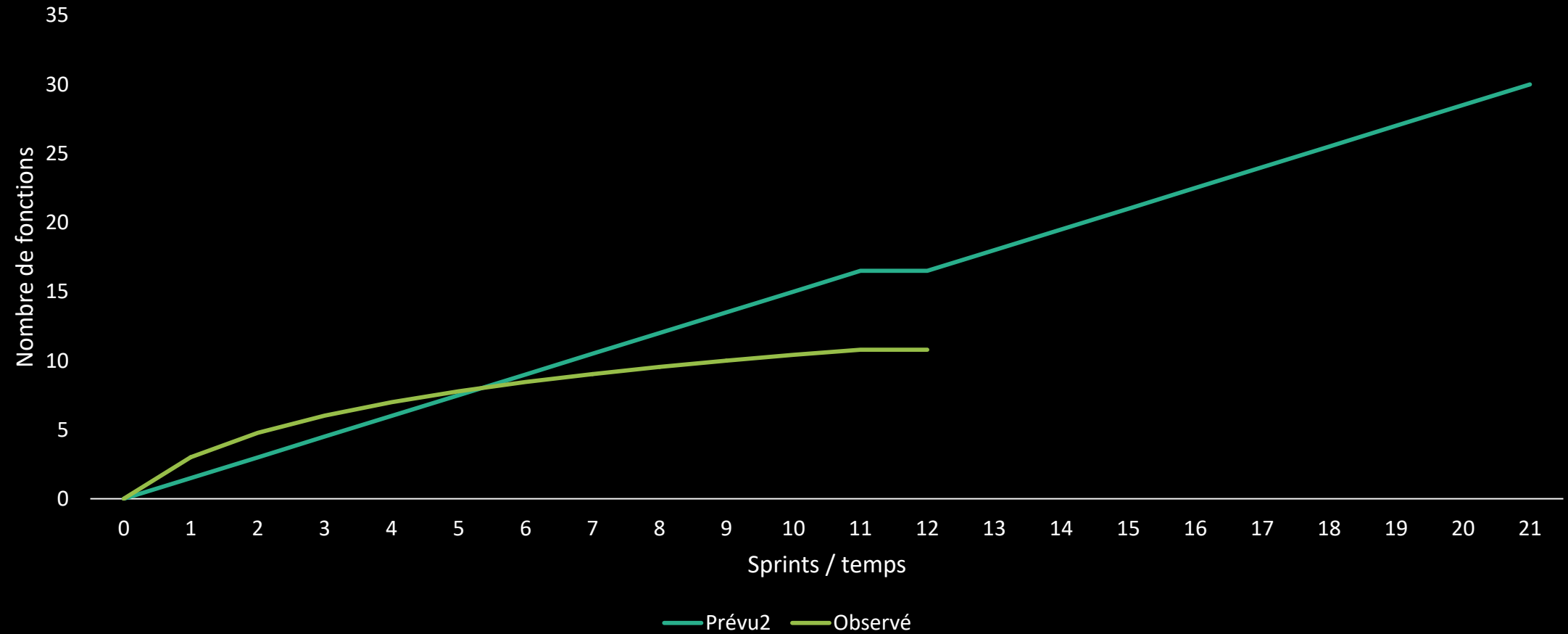
# Sprint(s) de nettoyage

Prendre des vacances pour revenir reposés



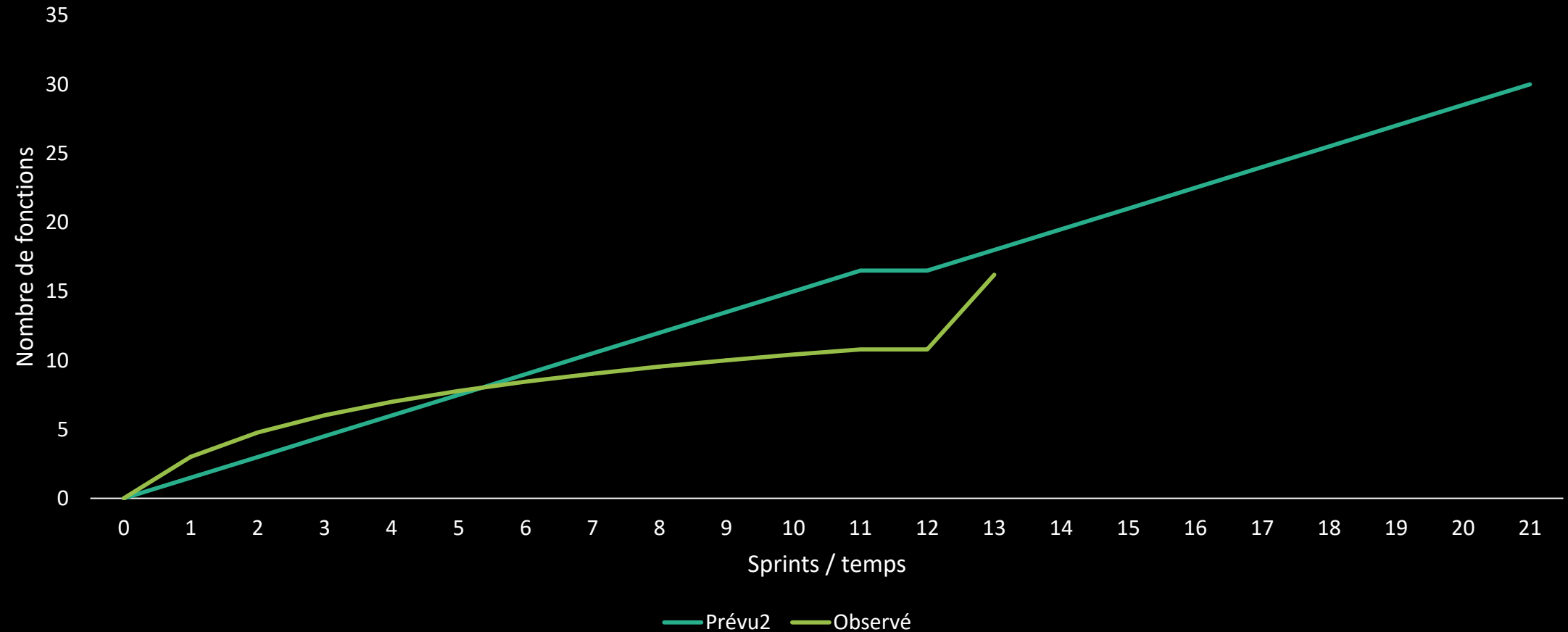
# Sprint(s) de nettoyage

Prendre des vacances pour revenir reposés



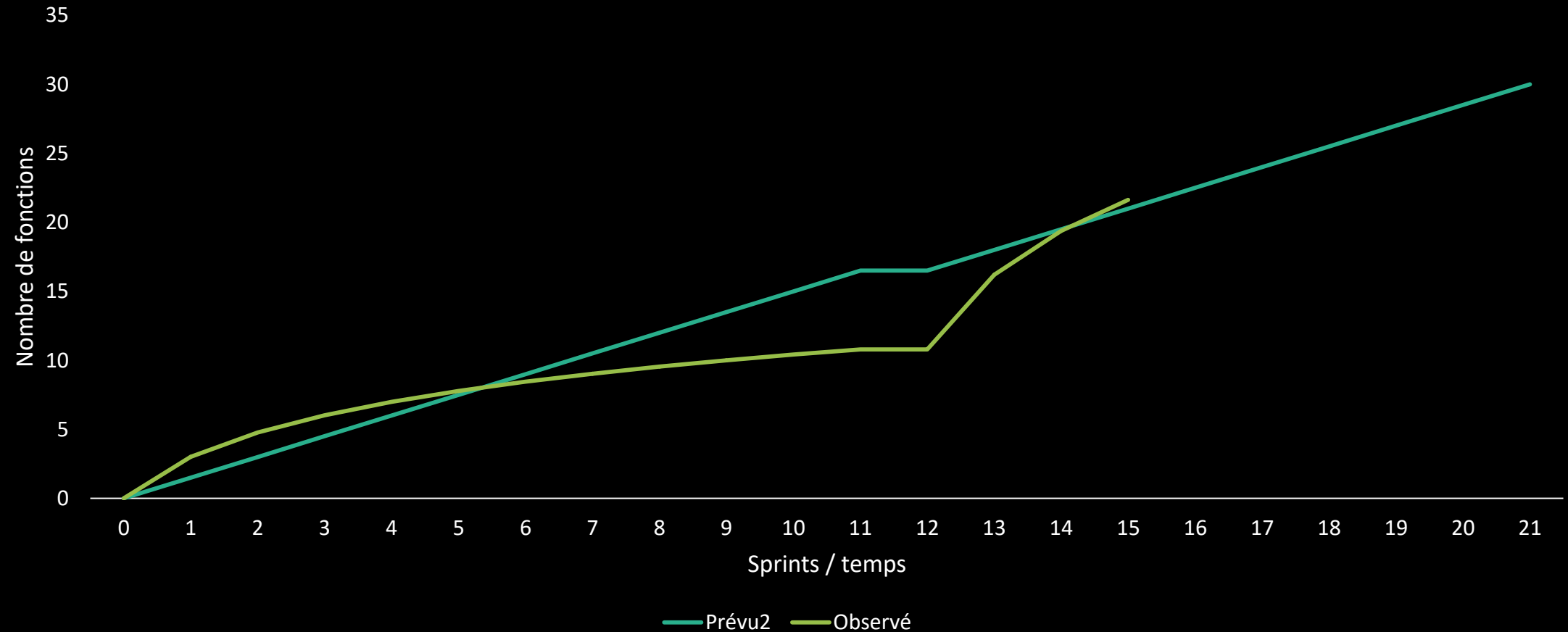
# Sprint(s) de nettoyage

Prendre des vacances pour revenir reposés



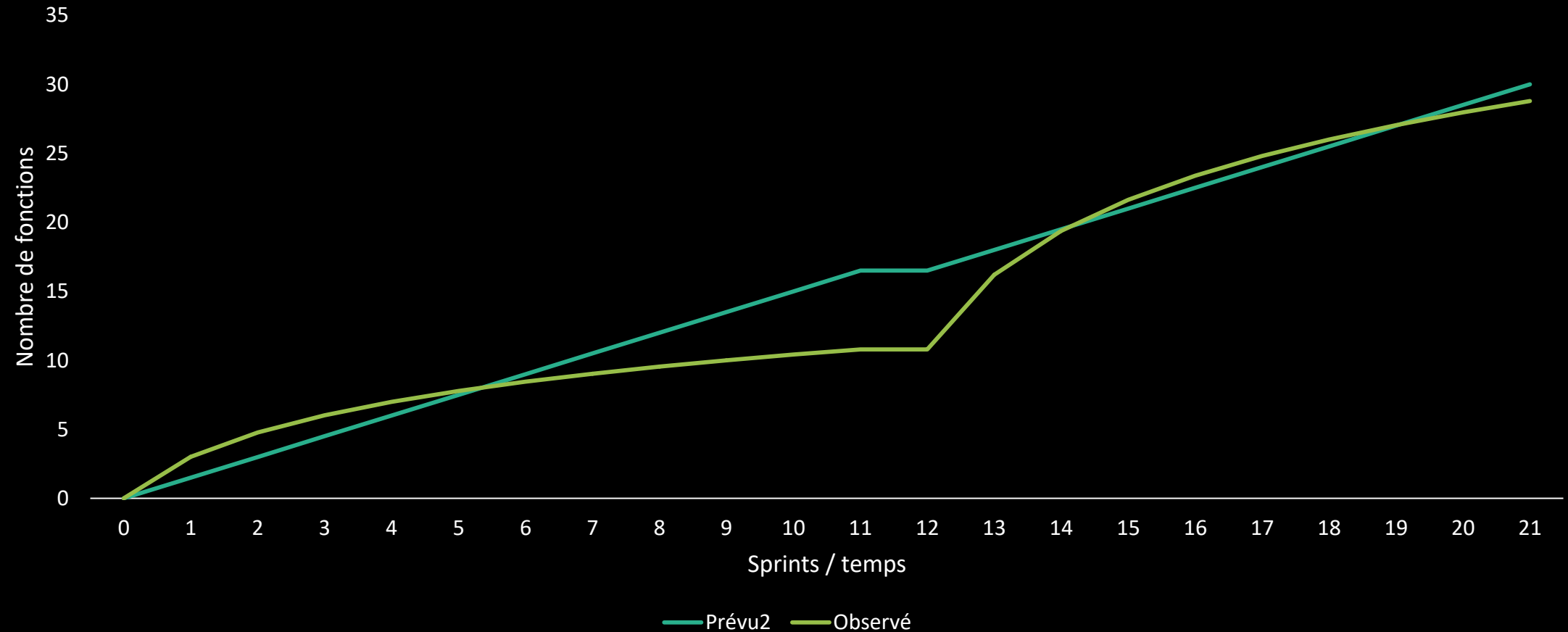
# Sprint(s) de nettoyage

Prendre des vacances pour revenir reposés



# Sprint(s) de nettoyage

Prendre des vacances pour revenir reposés





# Refactoring

Bonne solution

# Quoi ?

## Définition refactoring

**Modifier le code**

(ou de sa structure)

**Pour l'améliorer**

(plus facile à maintenir)

**Sans le modifier**

(comportement observable)

Qui ?

Celui qui touche au code

*« 4° A votre départ, veuillez à nettoyer l'endroit et à le laisser plus propre qu'à votre arrivée. »*

<https://www.sur-la-plage.com/articles/peut-on-faire-du-camping-a-la-plage-96.php>

Où ?

Partout, surtout main

*Branche principale*

*(trunk, master, main, ...)*

*Branche de fonctionnalité*

*(seulement si c'est court)*

Quand ?

Souvent oui

*« Si ça pue,  
changez-le. »*

La grand-mère de Kent Beck  
(À propos des enfants)

Quand ?  
Parfois, non

Yagni

(you ain't gonna need it)

3 strikes then out

(nettoyer la troisième fois)

# Comment ?

Un pas à la fois

## Préparer avec des tests

(rapides et faciles à lancer, tout le temps)

## Refactorer par itérations

(un petit changement à la fois)

## Un temps pour tout

(refactoring vs développement)

**ONE DOES NOT SIMPLY**

**EXPLAIN A TECHNICAL DEBT  
REMEDIATION PLAN TO ITS EXCO**