



# 07 Couverture

Partie 2 : conditions

Thibaut HENIN

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

Devs watching QA test the product

<https://www.youtube.com/watch?v=baY3SaIhf10>

Devs watching QA test the product



# Quel est le problème ?

La couverture de code était pas suffisante ?

# Exemple de test booléen

Quand freiner ?

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Vitesse > 100 ou pente > 30

ET

Mode = 1 ou 2

# Couverture des branchements

$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$

Toutes les instructions

(pas adapté)

Tous les chemins

(pas adapté)

Toutes les branches

(pas assez fin)

Critères de couverture

# Toutes les conditions

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

## Toutes les Conditions élémentaires

(un cas vrai, un cas faux)

$v > 100$	$p > 30$	$m = 1$	$m = 2$
1	1	1	0
0	0	0	1

# Toutes les conditions - décisions

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

## Toutes les Conditions élémentaires

(un cas vrai, un cas faux)

## L'expression globale

(un cas vrai, un cas faux)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	1	0
1	1	1	0	1

# Toutes les conditions multiples

$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$

Toutes les combinaisons

( $2^n$ , inutilisable)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Chaque condition qui modifie

(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Chaque condition qui modifie

(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$

Chaque condition qui modifie  
(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Chaque condition qui modifie

(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Chaque condition qui modifie

(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Toutes les conditions – décision modifiées

$$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$$

Chaque condition qui modifie

(deux tests où elle seule change)

$v > 100$	$p > 30$	$m = 1$	$m = 2$	global
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	???
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	???
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	???
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	???

# Au-delà de la couverture

Ce ne sont que des outils

# Modifier le code

$(v > 100 \vee p > 30) \wedge (m = 1 \vee m = 2)$

```
If (  
    ((v > 100) || (p > 30))  
    &&  
    ((m == 1) || (m == 2))  
    ) {  
    freiner() ;  
}
```

```
if (mode == 1 || mode == 2) {  
    if (v > 100 || p > 30) {  
        freiner() ;  
    }  
}
```

# L'instinct et l'expérience

Tout vient avec la pratique

Couverture = mesure

(pas un objectif)

Tester les erreurs courantes

(tout ce qui peut mal se passer)



**OUR TESTERS FOUND MORE BUGS**

**THAN OUR CUSTOMERS DID!**

imgflip.com

© <https://www.testbytes.net/blog/software-testing-memes/>