LA QUALITÉ
UNE ÉVIDENCE... VRAIMENT ?

AFUP Montpellier - 20 novembre 2025 - Smile

Julien Vinber

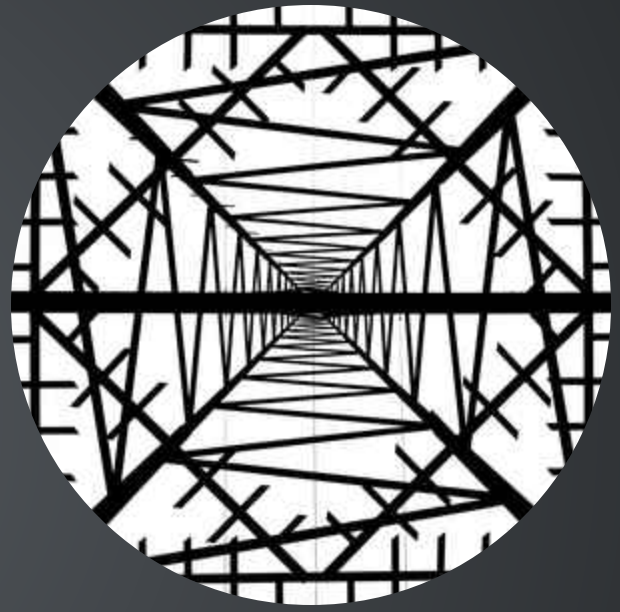
 PETIT SONDAGE

- Qui pense que la qualité, ce n'est pas important ?
- Qui a déjà entendu un collègue lui dire : "Haaaa ! C'est bon, ça marche"
- Qui pense que la qualité, ça coûte cher ?
- Pour vous, c'est quoi LA Qualité ?

JULIEN VINBER

- Junior depuis bientôt 25 ans
- Lead Dev / Architect
- Amoureux de PHP / Symfony

- **Recherche une nouvelle opportunité.**



@julienvinber

ESSAYONS DE DÉFINIR LA QUALITÉ.

WIKIPÉDIA

Dans le langage courant :

> *La qualité tend à désigner ce qui rend quelque chose supérieur à la moyenne.*
>
> — *Wikipédia*

WIKIPÉDIA

ISO 9000 :

L'« aptitude d'un ensemble de caractéristiques intrinsèques d'un objet (produit, service, ...) à satisfaire des exigences ». Dans ce contexte, le terme « qualité » peut être quelquefois utilisé avec des qualificatifs tels que médiocre, bon ou excellent.

— *Wikipédia*

OK… OUI… EUH… MAIS ENCORE.

Et si on demandait directement aux utilisateurs ?

PS : merci les LLM. (J'ai peut-être ajouté d'aller un peu dans la caricature.)

DÉVELOPPEUR PASSIONNÉ

"LA Qualité, c'est quand mon code est tellement propre et bien testé que je peux le relire six mois après en me disant 'putain, bien joué' au lieu de pleurer."

MAXIME LE "DÉVELOPPEUR"

"Haaaa ! C'est bon, ça marche..."

PO

"LA Qualité, c'est quand toutes les user stories passent en 'Done' avant la fin du sprint."

CLIENT

"LA Qualité, c'est quand ça fait exactement ce que je veux, que ça marche du premier coup, et que ça ne coûte pas plus cher que prévu."

LA SECRÉTAIRE À 2 ANS DE LA RETRAITE

"LA qualité, c'est quand ça marche comme avant."

PATRON D'UNE SOCIÉTÉ ÉDITRICE DE LOGICIEL

"LA qualité ? C'est quand mes clients renouvellent leur licence sans négocier."

MANAGER PAYÉ À LA PERFORMANCE

"LA Qualité, c'est livrer dans les temps et sous budget - peu importe ce qui se passe après."

ET DONC...

PREMIER ÉLÉMENT DE RÉPONSE :

- Il n'y a pas de définition universelle.
- Chaque partie définit LA Qualité par rapport à sa personnalité et ses préoccupations.

MA VISION

REVENONS À NOTRE SONDAGE.

Oui, vous avez tous raison. Mais...

Vous avez tous tort.

LA QUALITÉ N'EST PAS UN ÉTAT, MAIS UN PROCESSUS.

- L'immobilisme en informatique c'est reculer.
- On peut TOUJOURS faire mieux.
- Chercher à faire mieux qu'hier.

EXPLORONS QUELQUES PISTES D'AMÉLIORATION.

En vrac, c'est pour le plaisir.

⚠️ AVANT DE COMMENCER UN PETIT AVERTISSEMENT.

Oui. Je sais. Il y a des exceptions.

Mais attention à ne pas les transformer en règles.

LES TESTS

NON

ON VA PARLER DE COUVERTURE DE CODE

Ou plus exactement de la quantification de LA Qualité.

POURQUOI QUANTIFIER LA QUALITÉ ?

- Car les chiffres sont plus simples à manipuler.
- Car on doit communiquer.
- Car on travaille en équipe.

=> Mais c'est un piège.

COMMENT QUANTIFIER CE QUE L'ON A DÉJÀ DU MAL À DÉFINIR ?

- On se focalise sur quelques chiffres.
- Ce qui n'est pas quantifiable n'est pas pris en compte.
- On met des objectifs (70 % de couverture de code, 25 % de commentaires, nombre de jours sur de la dette...)

PETIT JEU : QUI EST LE MEILLEUR DEV ?

Voici une fonction.

```php
public function diviser(int $nb1, int $nb2): float
{
    try {
        echo ($nb1 / $nb2);
    } catch (DivisionByZeroError $e) {
        $this->logger->error($e);
        throw $e;
    }
}
```

Et on va juger les tests de deux dev.

UN DÉVELOPPEUR A.

```php
public function testDiviser()
{
    $this->assertEquals(2, $this->math->diviser(10, 5));
    $this->assertEquals(-2, $this->math->diviser(10, -5));
    $this->assertEquals(2, $this->math->diviser(-10, -5));
    $this->assertEquals(0, $this->math->diviser(0, 10));
}
```

MAXIME LE DÉVELOPPEUR.

```php
public function testDiviser()
{
    $this->expectException(\Throwable::class);
    diviser(10, 0);
}
```

LE MEILLEUR EST BIEN SÛR :

Maxime (couverture de code à 100 % et code plus vite)

Et pour le plaisir

```php
declare(strict_types=1)

public function diviser(string $nb1, int $nb2): float
{
}
```

Allume vert les tests...

CONCLUSION

Je comprends pourquoi on cherche à quantifier.

Mais c'est loin d'être une bonne idée.

=> Ça pousse les mauvais à faire encore plus de merde.

VISER LE 100 %.

Je ne suis pas à une contradiction près.

Visez pour VOUS le 100 %.

POURQUOI PAS 99 % ?

- On doit se poser la question de savoir si oui ou non on le fait.

=>99 % coûte plus cher que 100 %.

- On risque beaucoup plus d'oublier.
- On doit passer par des phases de rattrapage.

ALORS QUE 100 %

- Pas besoin de réfléchir : on le fait.
- À force, ça devient un réflexe.
- Et si en plus on cherche à comprendre : on apprend.

OUI MAIS...

Non, pas d'excuse.

Réfléchissez et ignorez...

Tous les outils qui appliquent des règles permettent d'ignorer.

EXEMPLE PHPUNIT...

```php
class Demonstration {
    /**
     * @codeCoverageIgnore
     */
    public function ignorerLaMethode() {
        $config = 'legacy_config';
        $this->connect($config);
    }

    public function ignorerDesLignes($valeur) {
        if ($valeur < 0) { return false; }

        // @codeCoverageIgnoreStart
        if (getenv('ENV') === 'dev') {var_dump($valeur);}
        // @codeCoverageIgnoreEnd
        return $valeur * 2;
    }
}
```

Doc PhpUnit

UN PETIT CONSEIL :

- Un pro utilise un vrai IDE.
- Non, VSCode, c'est un jouet, pas un outil pour les pros.
- PHPStorm + des extensions
  - Php Inspections (EA Extended)
  - PHP Hammer
  - Symfony Plugin
  - PHP Annotations
  - Twig
- Soyez psychorigide sur les alertes.

DEMANDER

- À votre Lead.
- À votre CTO.
- Au client.
- Au PO.

- Ne leur faites pas non plus perdre du temps.

- Reformulez les US.
- Remettez en cause l'US.
- Faites des propositions.
- Validez l'US.
- Comprenez la demande.
- Comprenez le métier.

Si l'entreprise vous demande de ne pas gêner les autres :

Alors c'est elle qui n'a rien compris.

CE N'EST PAS PARCE QUE ÇA MARCHE QUE ÇA MARCHE.

???

PRENONS UN EXEMPLE

```php
function maFonction($maVariable)
{
    if(!$maVariable) {
        return false;
    }

    return 10/$maVariable;
}
```

IL FAUT :

- Typer.
- Nommer.
- Éviter toute ambiguïté.
- Éviter la magie.
- Éviter de vouloir gagner du temps et des lignes de code.

SOLID

Retenez a minima :

- **S**ingle Responsability
- **D**ependency Inversion

KISS

keep it simple, stupid, « garde-le simple, idiot »

Il est simple de faire compliqué, mais compliqué de faire simple.

ET SI ON A ENCORE UN PEU DE TEMPS

- La veille...
- Les meetups... (+6 %)
- Il n'y a pas qu'une seule solution.
- Toujours réfléchir et se mettre à la place des autres.