

# Qualité logiciel & tests

My Digital School - Bachelor 3 dev 2019 - Louis Beltramo



# Présentations

# Présentations

Louis Beltramo - [louis@beltramo.me](mailto:louis@beltramo.me)

Freelance

Développeur d'applications web et mobiles

Spécialisé en Front-End

# Présentations

A vous !

# Au programme

- Un peu de théorie
- Mise en place de tests unitaires (PHPUnit, Jest)
- Test Driven Development (TDD)
- Mise en place de tests d'acceptation automatisés (Codeception, Cypress)
- Behavior Driven Development (BDD)

# Qualité logiciel

C'est quoi ?

Quels éléments à prendre en compte ?

Comment définir cette qualité ?

Comment évaluer cette qualité ?

# Qualité logiciel

niveau de qualité

Assurer un niveau de qualité défini  
en accord **avec le client**

Criticité du logiciel/service

Maintenabilité

Transmission

# Tests

Vérifier que le logiciel est conforme aux attentes

Ensemble de cas à tester (état de l'objet à tester avant exécution du test, actions ou données en entrée, valeurs ou observations attendues, et état de l'objet après exécution), éventuellement accompagné d'une procédure d'exécution(séquence d'actions à exécuter). Il est lié à un objectif.



# Pourquoi tester ?

Prévenir les bugs

**Garantir** le niveau de qualité

**Assurer** la maintenabilité

**Améliorer** la transmission et la compréhension par les autres (ou vous même plus tard)

# Pourquoi ne pas tester ?

Trop de temps et d'efforts

Pas le temps

Pas de ROI, difficile à justifier

On en peut pas tout tester

Je veux développer des fonctionnalités, pas des tests !

Une appli est  
toujours testée...

...par les  
utilisateurs !

Comment tester ?

# Mise en place progressive

Amélioration des pratiques

- Pas de test
- Test manuel
- Tests automatiques
- C.I. (intégration continue)
- Chaos Monkey

# Comment tester ?

- Que tester ?
- Quand tester ?
- Combien de tests ?
- Tester à quel point ?

Juste ce qu'il faut



# Niveaux de tests

- Tests unitaires
- Tests d'intégration
- Tests système
- Tests d'acceptation

# Tests unitaires

## Unit Test

“procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un logiciel”

# Tests d'intégration

“chacun des modules indépendants  
du logiciel est assemblé et testé  
dans l'ensemble”

# Tests systèmes

“Les tests système prennent comme entrée tous les composants logiciels « intégrés »”

“Les tests système vérifient non seulement la conception, mais également le comportement et les attentes présumées du client.”

utilisabilité, performance, charge, régression

# Tests d'acceptation

Recette client

Nécessite un plan de test, un  
procédure précise

# Concrètement, pour une application web

1. Tests statiques : linter, typage statique
2. Tests unitaires : au niveau des fonctions et modules/classes
3. Tests d'intégration : vérification de plusieurs modules dépendants
4. Tests End-to-End (e2e) : ensemble de parcours du site automatisé
5. Tests systèmes : au besoin, tester les performances, montée en charge ou encore la sécurité avec des tests d'intrusions
6. Tests d'acceptation : produire avec le client une procédure de recette pour permettre de valider les livraisons

# Tests unitaires

# Un test unitaire

on test en isolement

C'est

- déterministe
- indépendant

Ce n'est pas

- exhaustif



# Comment isoler ?

Avec les mocks (“bouchons”)

Simuler les opérations “extérieures”  
:

- envoi d’un email
- accès BDD
- appel d’API
- ...

# Quoi tester ?

- Le cas nominal
- Les cas d'erreurs
- Les cas aux limites