

Le “Clean Code” appliqué aux tests auto



Agenda

- ◆ Contexte de travail
- ◆ Rappel et définition du Clean Code
- ◆ Recommandations
- ◆ Retour d'expérience
- ◆ Bonus





Rija William RALITERA

Ingénieur Réseaux et Télécommunications: certifié Cisco & VMware

Ingénieur Qualité Logiciel: Stratégie / Automatisation depuis 2015

Apprenti Blogueur / Speaker

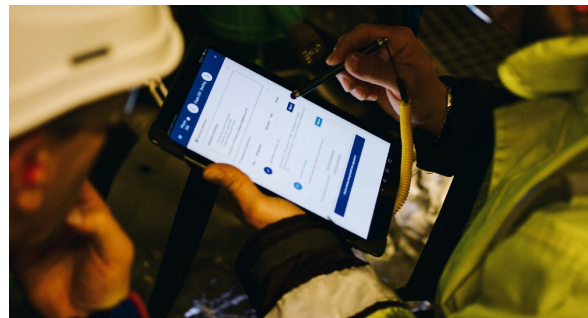
Perso



Contexte de travail

- Poste: Lead QA Chez SITEFLOW

- 32 Techs
- 8 QAs

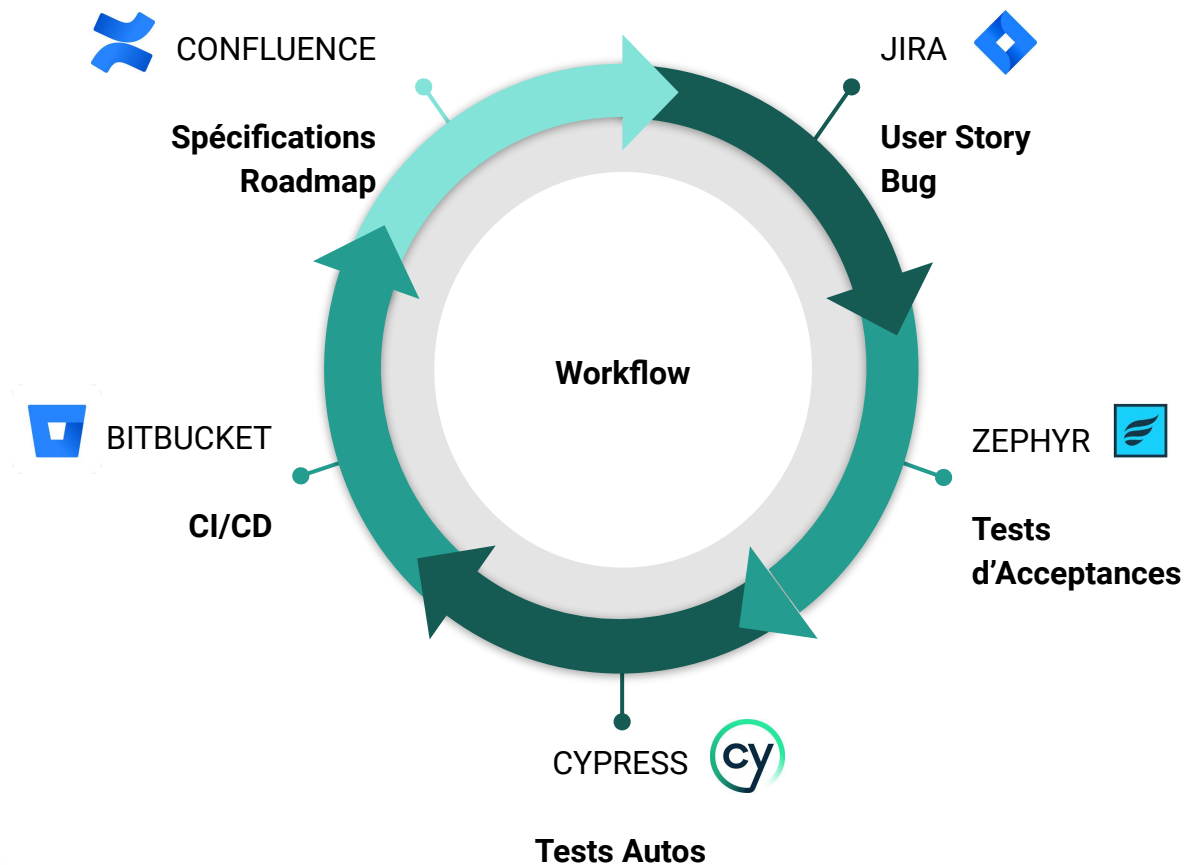


- SITEFLOW

- Créé en 2017
- Transformation Digitale des Industries Complexes
- + 60 employés
- Certifié ISO27001 et 9001



Contexte de travail



Rappel: Test autos

- ◆ Projet de développement à part entière
- ◆ Évolue avec le produit
- ◆ Travail en équipe
- ◆ Documentation
- ◆ Revue de code
- ◆ Maintenance du code
- ◆ Bugs créés par le Framework ou le code de test



Définition du Clean Code

- ◆ D'après Robert C. Martin (uncle BOB) auteur du livre: Clean Code: A Handbook of Agile Software Craftsmanship:
 - “Le code est propre s'il peut être **compris facilement** - par tous les membres de l'équipe.
 - Un code propre **peut être lu et amélioré** par un développeur autre que son auteur initial.
 - Quand un code est **compréhensible** , il devient **lisible, modifiable, extensible** et **maintenable**.”

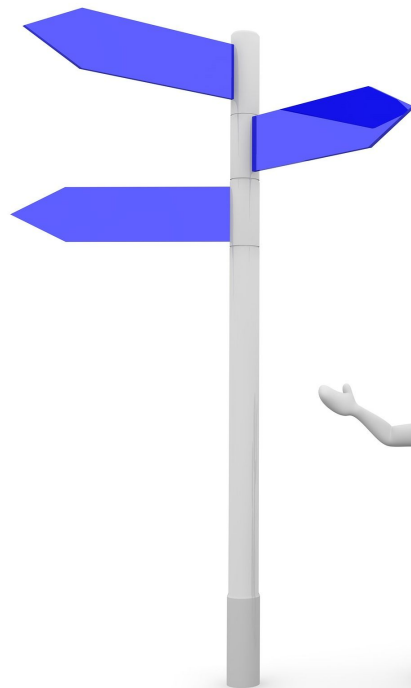


Indicateurs de mauvais code

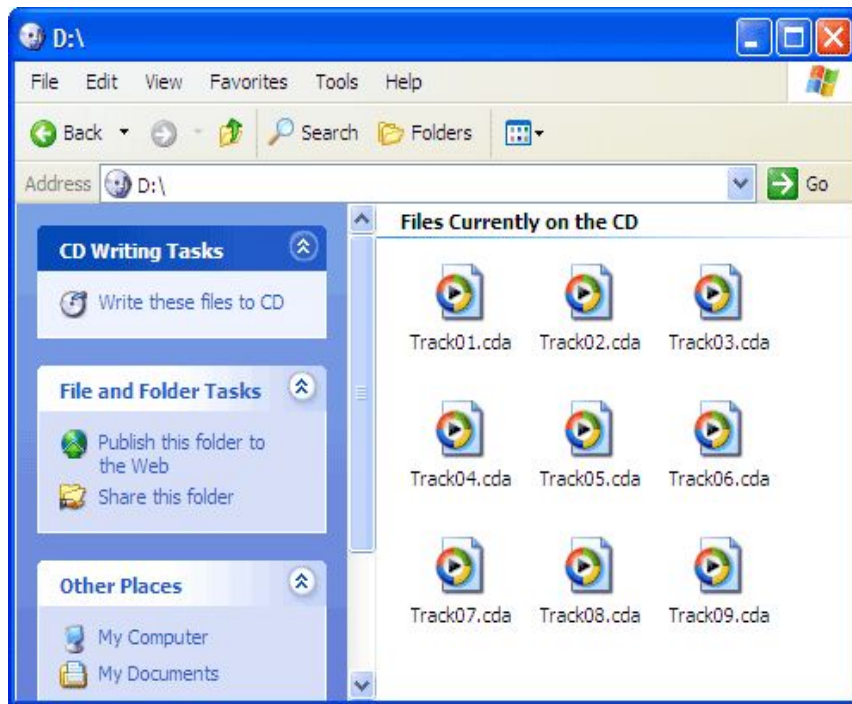
- ◆ **Scalabilité** : plus on a de scénarios de tests, plus c'est difficile à gérer
- ◆ **Fragilité** : un seul changement impact plusieurs tests
- ◆ **Complexité**: implémentation incompréhensible
- ◆ **Répétition**: code de test dupliqué à plusieurs endroits



Clause de non-responsabilité



Le bon vieux temps ...



❖ R1: Gestion de l'arborescence

✓  integration

✓  performanceandsecuritytest

 performanceAndsecuritytest.spec.ts

 createUser.spec.ts

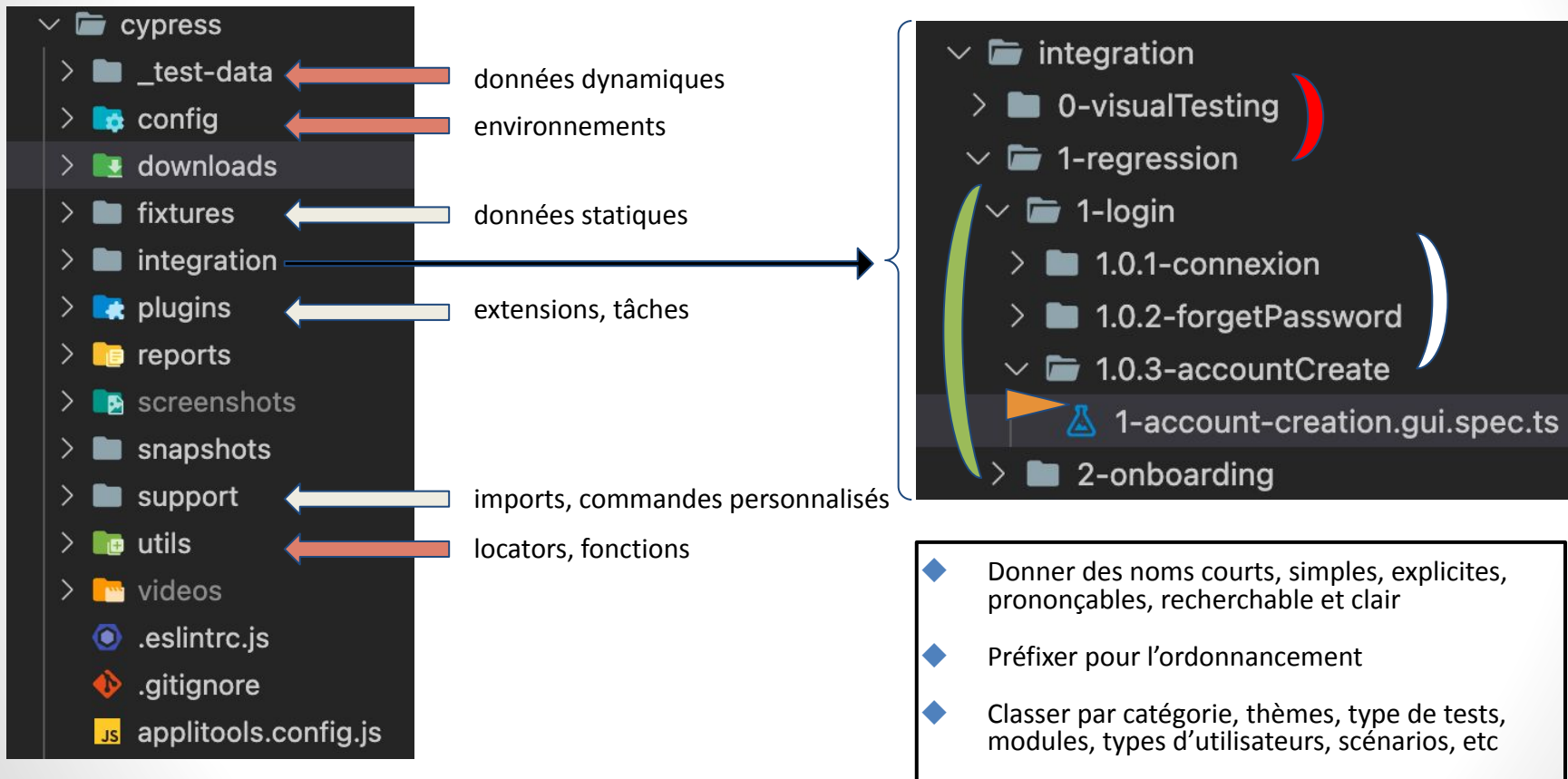
 forgetPassword.spec.ts

 onboardingtest.spec.ts

 visual_testing.spec.ts



❖ R1: Gestion de l'arborescence





R2: Gestion des noms

- Trop longue
- Non significatif
- Ordre args
- Nommage hétérogène
- Non recherachable
- Générique

```
Add only
describe('This test will test how to download upload files into the application after login to the application', () => {
  before(() => {
    window.localStorage.setItem('currentLang', 'fr');
  });

  beforeEach(() => {
    cy.saveLocalStorage();
    cy.visit('auth');
  });

  function func1(pass:string,user:string) {
    cy.get('.mail').type(user);
    cy.get('.password').type(pass);
  }

  const a = 'mail@gmail.com';
  const very_securedPassword_Userfor_Scenario1 = '12345';
  const testurl = Cypress.config('baseUrl');

  // Scénario 6 : Succès
  Open Cypress | Set "only" | Add only
  it('scenario 1', () => {
    func1(very_securedPassword_Userfor_Scenario1, a);
    cy.visit(`${testurl}/equipments`);
    cy.get('.sfx-button').eq(0).click();
    cy.get('[class="upload-file"][type="file"]').attachFile({ filePath: 'test.xlsx', encoding: 'utf-8' });
  });

  Open Cypress | Set "only" | Add only
  it('scenario 2', () => {
  });
});
```





R2: Gestion des noms

- Courtes
- Simples
- Explicites
- Prononçables
- Claires

- camelCase
- under_score

- Termes métiers
- Termes techniques
- Personas
- Recherchable

- Descriptifs
- Structuré
- Mots clés (NOT, lender)

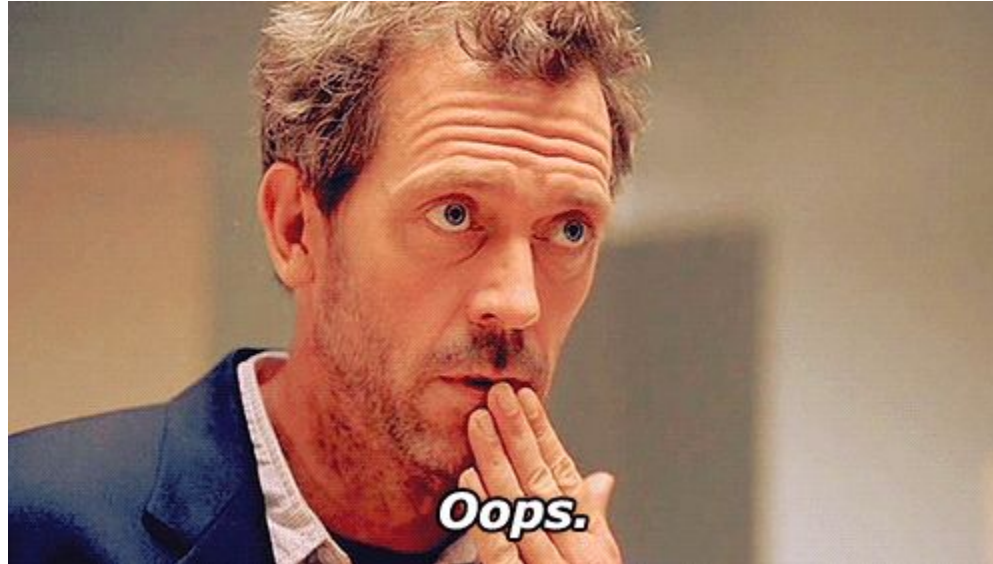
Add only

```
describe('upload testing', () => {  
  before(() => {  
    window.localStorage.setItem('currentLang', 'fr');  
  });  
  
  beforeEach(() => {  
    cy.saveLocalStorage();  
    cy.visit('auth');  
  });  
  
  function loginViaGui(mail: string, password: string) {  
    cy.get('.mail').type(mail);  
    cy.get('.password').type(password);  
  }  
  
  const lenderUser = 'mail@gmail.com';  
  const lenderPassword = '12345';  
  const equipmentsUrl = `${Cypress.config('baseUrl')}/equipments`;  
  
  // Scénario 6 : Succès  
  Open Cypress | Set "only" | Add only  
  it('lender user should be able to upload a xlsx file', () => {  
    loginViaGui(lenderUser, lenderPassword);  
    cy.visit(equipmentsUrl);  
    cy.get('.sfx-button').eq(0).click();  
    cy.get('[class="upload-file"][type="file"]').attachFile({ filePath: 'test.xlsx', encoding: 'utf-8' });  
  });  
  
  Open Cypress | Set "only" | Add only  
  it('lender user should NOT be able to upload a pdf file', () => {  
  });  
});
```



❖ R3: Gestion des logs et des erreurs

Les QA vont générer des bugs !!!!!



❖ R3: Gestion des logs et des erreurs

```
function loginViaGui(mail: string, password: string) {  
  cy.get('.mail').type(mail);  
  cy.get('.password').type(password);  
  cy.log(mail, password);  
}
```

déroulement, historique, debug

```
function uploadFile(fileName: string) {  
  if(fs.existsSync(fileName)) {  
    cy.get('[class="upload-file"][type="file"]').attachFile({ filePath: 'test.xlsx', encoding: 'utf-8' });  
  }  
  
  cy.log("file does not exist");  
  return null;  
}
```

traitement d'erreurs identifiées

```
// Scénario 6 : Succès  
Open Cypress | Set "only"  
it('lender user should be able to upload a xlsx file', () => {  
  loginViaGui(lenderUser, lenderPassword);  
  cy.visit(equipmentsUrl);  
  cy.get('.sfx-button').eq(0).click();  
  uploadFile('test.xlsx');  
});
```



❖ R4: Gestion des commentaires

- Informations inutiles: historique, initiateurs...

- Excès de commentaires par ligne de code
- Non uniformes
- Redondants

- Commentaires de début/fin de blocks
- Séparateurs de codes inutiles

```
/*
Initial writer: Nico Robin
This function is to login on the main page
It is a version 2 because we will create a loginViaAPI function */
function loginViaGui(mail: string, password: string) {
  cy.get('.mail').type(mail); // on tape le mail de l'utilisateur
  cy.get('.password').type(password); // we type the password
  cy.log(mail, password); // on affiche le mail et password
  cy.url().should("/home") // we check that we are on the main page
}

/***** Start Scenario 992*****/
Open Cypress | Set ".only"
it('lender user should be able to upload a xlsx file', () => {
  loginViaGui(lenderUser, lenderPassword);
  cy.visit(equipmentsUrl);
  cy.get('.sfx-button').eq(0).click();
  uploadFile('test.xlsx');
});
/***** End Scenario 992*****/

/***** Start Scenario 993*****/
Open Cypress | Set ".only"
it('lender user should NOT be able to upload a pdf file', () => {
});
/***** End Scenario 993*****/
});
```



❖ R4: Gestion des commentaires

- jsDoc
 - Description rapide
 - Paramètres
 - Exemple
-
- Problèmes identifiés
 - A faire
 - Avertissements
 - Tickets JIRA

```
/**
 * login function via Gui
 * @param mail
 * @param password
 */
function loginViaGui(mail: string, password: string) {
  cy.get('.mail').type(mail);
  cy.get('.password').type(password);
  cy.log(mail, password);
  cy.url().should("/home"); // FIXME: It failed because of a popup, add a popup skip
  // TODO: Add more checks like username is visible
}
```

```
(local function) loginViaGui(mail: string, password: string): void
login function via Gui
/*
Op
it @param mail
   @param password
loginViaGui(lenderUser, lenderPassword);
cy.visit(equipmentsUrl);
```



❖ R5: Factorisation de code

- Code dupliqué

Open Cypress | Set ".only"

```
it('lender user should be able to upload a XLSX file', () => {  
  cy.get('.mail').type(lenderUser);  
  cy.get('.password').type(lenderPassword);  
  cy.log(lenderUser, lenderPassword);  
  cy.url().should('contain', '/home');  
  
  cy.visit(equipmentsUrl);  
  cy.get('.sfx-button').eq(0).click();  
  uploadFile('test.xlsx');  
  cy.get('[data-cy="message"]').should('contain', "file uploaded")  
});
```

Open Cypress | Set ".only"

```
it('lender user should NOT be able to upload a PDF file', () => {  
  cy.get('.mail').type(lenderUser);  
  cy.get('.password').type(lenderPassword);  
  cy.log(lenderUser, lenderPassword);  
  cy.url().should('contain', '/home');  
  
  cy.visit(equipmentsUrl);  
  cy.get('.sfx-button').eq(0).click();  
  uploadFile('test.pdf');  
  cy.get('[data-cy="message"]').should('contain', "Failed upload")  
});
```





R5: Factorisation de code

- Fonction locale
- Arguments limités
- Optimisation contrôlée
- Code factorisé
- Gain de lignes de codes
- Gain en lisibilité

```
function loginViaGui(mail: string, password: string) {  
  cy.get('.mail').type(mail);  
  cy.get('.password').type(password);  
  cy.log(mail, password);  
  cy.url().should('contain', '/home');  
}
```

Open Cypress | Set "only"

```
it('lender user should be able to upload a XLSX file', () => {  
  loginViaGui(lenderUser, lenderPassword);  
  cy.visit(equipmentsUrl);  
  cy.get('.sfx-button').eq(0).click();  
  uploadFile('test.xlsx');  
  cy.get('[data-cy="message"]').should('contain', "file uploaded")  
});
```

Open Cypress | Set "only"

```
it('lender user should NOT be able to upload a PDF file', () => {  
  loginViaGui(lenderUser, lenderPassword);  
  cy.visit(equipmentsUrl);  
  cy.get('.sfx-button').eq(0).click();  
  uploadFile('test.pdf');  
  cy.get('[data-cy="message"]').should('contain', "Failed upload")  
});  
});
```





R5: Factorisation de code

- Fonction globale simple
- Amélioration continue
- Revue de code
- Fonction locale supprimée

```
Cypress.Commands.add('loginViaGui', (mail: string, password: string) => {  
  cy.session([mail, password], () => {  
    cy.get('.mail').type(mail);  
    cy.get('.password').type(password);  
    cy.log(mail, password);  
    cy.url().should('contain', '/home');  
  });  
});
```

- Fonction appelée une seule fois
- Fonction utilisable partout

```
beforeEach(() => {  
  cy.loginViaGui(lenderUser, lenderPassword);  
});  
  
Open Cypress | Set ".only"  
it('lender user should be able to upload a XLSX file', () => {  
  cy.visit(equipmentsUrl);  
  // ...  
});  
  
Open Cypress | Set ".only"  
it('lender user should NOT be able to upload a PDF file', () => {  
  cy.visit(equipmentsUrl);  
  // ...  
});  
});
```





R6: Le formatage de code

```
^
Add only
describe('upload testing', () => {
const lender_user = 'mail@gmail.com';
const lenderPasswr = '12345';
const equipmentsUrl = `${Cypress.config('baseUrl')}/equipments`;
```

```
^
^
^
^
function uploadFile(fileName: string) {
if(fs.existsSync(fileName)) {
    cy.get('[class="upload-file"][type="file"]').attachFile({ filePath: 'test.xlsx', encoding: 'utf-8' })
}
cy.log("file does not exist");
return null;
}
```

- Identifier 'lender_user' is not in camel case. eslint
- "Passwr": Unknown word.cSpell
- More than 1 blank line not allowed.eslint
- Expected indentation of 4 spaces but found 0.eslint
- Insert `;`eslint
- Strings must use singlequote.eslint
- Expected blank line before this statement.eslint



❖ R6: Le formatage de code

```
describe('upload testing', () => {  
  const lenderUser = 'mail@gmail.com';  
  const lenderPassword = '12345';  
  const equipmentsUrl = `${Cypress.config('baseUrl')}/equipments`;  
  
  function uploadFile(fileName: string) {  
    if (fs.existsSync(fileName)) {  
      cy.get('[class="upload-file"][type="file"]').attachFile({ filePath: 'test.xlsx', encoding: 'utf-8' });  
    }  
  
    cy.log('file does not exist');  
  
    return null;  
  }  
}
```

- ◆ Créer des règles pour tout le monde
- ◆ Utiliser des linters pour vérifier l'application des règles
- ◆ Appliquer **INTELLIGEMMENT** les règles



Retour d'expérience

- ◆ Revue de code plus vite et plus fluide
- ◆ Uniformisation dans la manière de coder les scénarios
- ◆ Réutilisabilité des fonctions
- ◆ Gain en productivité
- ◆ Fiabilité et robustesse des tests
- ◆ Apprentissage plus vite des nouveaux QA
- ◆ Processus et méthodologie de travail uniformisés
- ◆ Renforcement de l'état d'esprit (FAIRE DE LA QUALITÉ)
- ◆ Gestion des runs plus facile grâce à la structure
- ◆ Amélioration en continue



Retour d'expérience

1 an avant

Latest runs

FILTER BY

 All Time  Status  Branch  Committer  Tag  Flaky Tests

 Merge remote-tracking branch 'origin/develop' into feature/FEATURE-6998-test-aut...
  Ran a year ago ·  01:41:26 ·  feature/FEATURE-6998-test-auto-projects-mod... ·  Jenkins 6 · # 7

 0  1  133  61

22/11/2021

Latest runs

FILTER BY

Last updated: 10:46:46pm 

 Nov 22, 2021  Status  release/release-1.9.60.7  Committer  Release1  Flaky Tests

 [WB][RELEASE-]: Update version: 1.9.60.7.53 
  Ran 23 hours ago ·  01:04:23 ·  release/release-1.9.60.7 ·  Bitbucket 44269 · # 2980

 0  0  892  101



Bonus: Plugins Visual Studio Code



Code Spell Checker
Spelling checker for source code
Street Side Software



ES6 Mocha Snippets
Shortcuts to reduce the
Cory Noonan



ESLint
Integrates ESLint
Microsoft



Prettier - Code formatter
Code formatter using prettier
Prettier



Trailing Spaces
Highlight trailing spaces
Shardul Mahadik



TODO Highlight
highlight TODOs, FIXMEs,
Wayou Liu



Todo Tree
Show TODO, FIXME, etc. d
Gruntfuggly

- ◆ Facilitateur du quotidien
- ◆ Augmentation de productivité
- ◆ Apprentissage en continue
- ◆ Amélioration de la qualité



Résumé

1	Arborescence et noms	• Utiliser des noms courts, simples, explicites, prononçables, recherchable et clair • utiliser un système de nommage: camelCase, under_score
2	Logs et erreurs	• Loguer les valeurs importantes • Traiter les erreurs connues
3	Commentaires	• Utiliser le moins de commentaires possible • Utiliser du jsDoc • Utiliser des mots clés TODO, FIXME
4	Factorisation	• Limiter les codes dupliqués • Utiliser les fonctions avancés de votre framework de test: custom command
5	Formatage	• Mettre en place des règles • Essayer de suivre les règles
6	Plugins	• Utiliser des plugins dédiés pour vous aider • Appliquer INTELLIGEMENT les règles données



Merci!

Que chacun balaie devant son ordinateur
et les codes seront nettes.

[@RaliteraW](#)

<https://rija-william-ralitera.medium.com>

<https://www.linkedin.com/in/rija-william-ralitera>

