



Soirée du  
test logiciel

Conférence

# Arrêtons de (dé)tester

nos applications et nos utilisateurs



Stanley SERVICAL



Louis Frédice  
NJAKO MOLOM



# Problématiques abordées

## 1. Automatisation des tests E2E

- Pour répondre au TTM
- Seul véritable moyen pour valider les scénarios métiers
- Comment simplifier les tests E2E ?

## 2. Assurer l'Accessibilité Numérique

- Trop souvent négligée
- Plus réactif que proactif
- Comment garantir l'accessibilité numérique ?



Soirée du  
test logiciel

Conférence



**Architecte Logiciel**  
**Coach & Animateur Craft**



**Architecte Logiciel**  
**Coach & Animateur Craft**

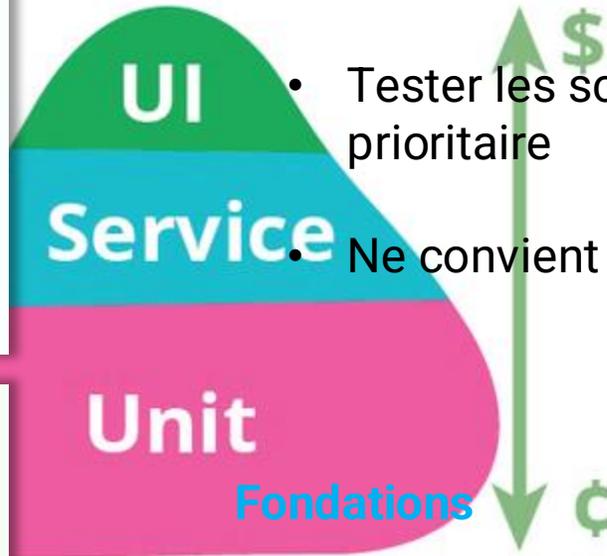
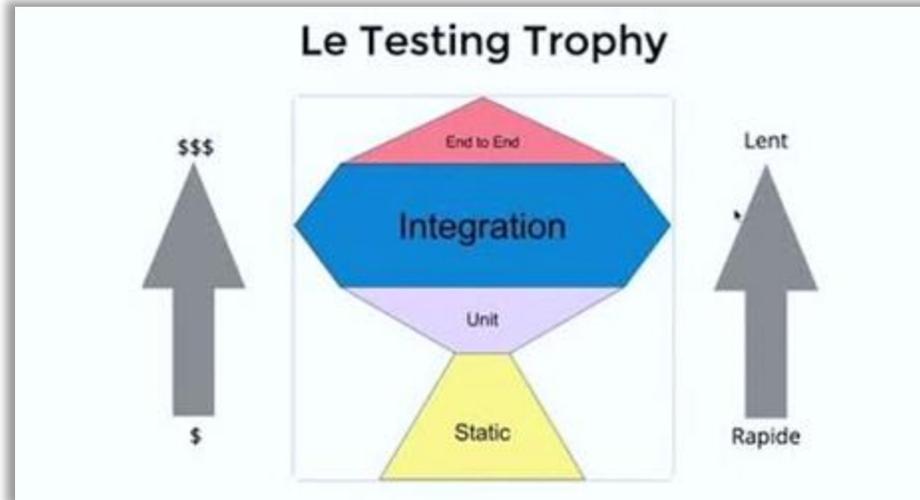


# #1

## Comment tester ?

# Tester oui, mais quoi ?

## Quelques limites



- Le nombre de tests est considéré à tort comme un indicateur de performance clé.

- Tester les scénarios utilisateurs réels n'est pas prioritaire

- Ne convient pas à tous les contextes

- Plus un test est de haut niveau, plus il est lent

- Plus un test est de haut niveau, plus il est coûteux



## Quels enseignements ?

- 1. Bien réfléchir avant de choisir votre modèle :  
Pyramide de tests, Hexagone, Testing Matrix**
- 2. Le seul test qui puisse véritablement valider  
l'utilisabilité de l'application est un test E2E**



# Le constat sur les tests E2E

## 1. Priorité

- Souvent en bas de la chaîne de valeur par rapports aux nouvelles fonctionnalités
- Souvent considérés comme moins importants que les TU ou TI par les développeurs

## 2. Complexité fonctionnelle

Plusieurs systèmes, interface utilisateurs

## 3. Complexité technique

- Nécessite parfois des outils spécifiques pour les systèmes complexes
- Choix et configuration des outils difficiles
- Ecriture des scénarios

## 4. Temps et coûts

Long à exécuter, ressources importantes



# #2

## Test E2E idéal



# Le constat sur les tests E2E

## 1. Automatisable

## 2. Représenter des vrais

## 3. Facile à écrire, à maintenir

- Exprimer les scénarios

## 4. Interagir comme un utilisateur

```
import { verifyChildByTestId } from "../testHelper";

describe('Weather', () => {
  it('should be display nothing when no town is selected', () => {
    cy.visit('/');
    cy.getByTestId("start-button").click();

    cy.getByTestId("weather-details-empty").should('have.text', 'Nothing to display');
  })

  it('should display correct town list', () => {
    cy.visit('/');
    cy.getByTestId("start-button").click();

    verifyChildByTestId("available-towns", ['Douala', 'Tunis', 'Limoges']);
  })
});
```



Soirée du  
test logiciel

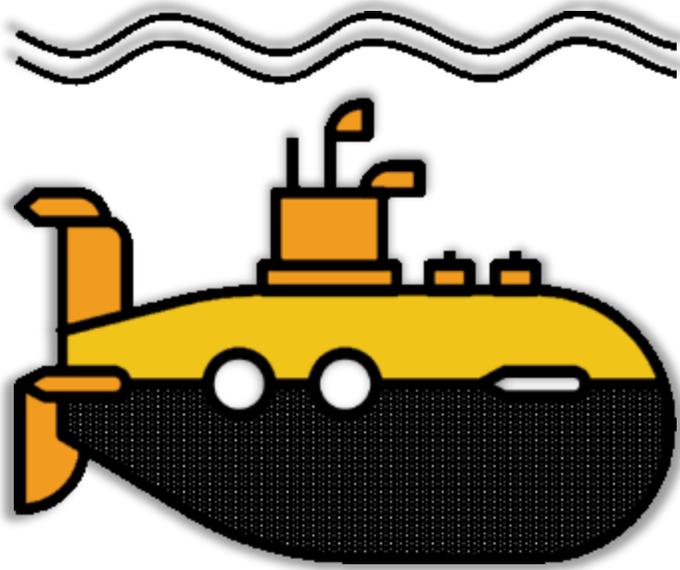
Conférence

# #3

## Solution



U  
U  
V



User-centric Usecases Validator

OpenSource

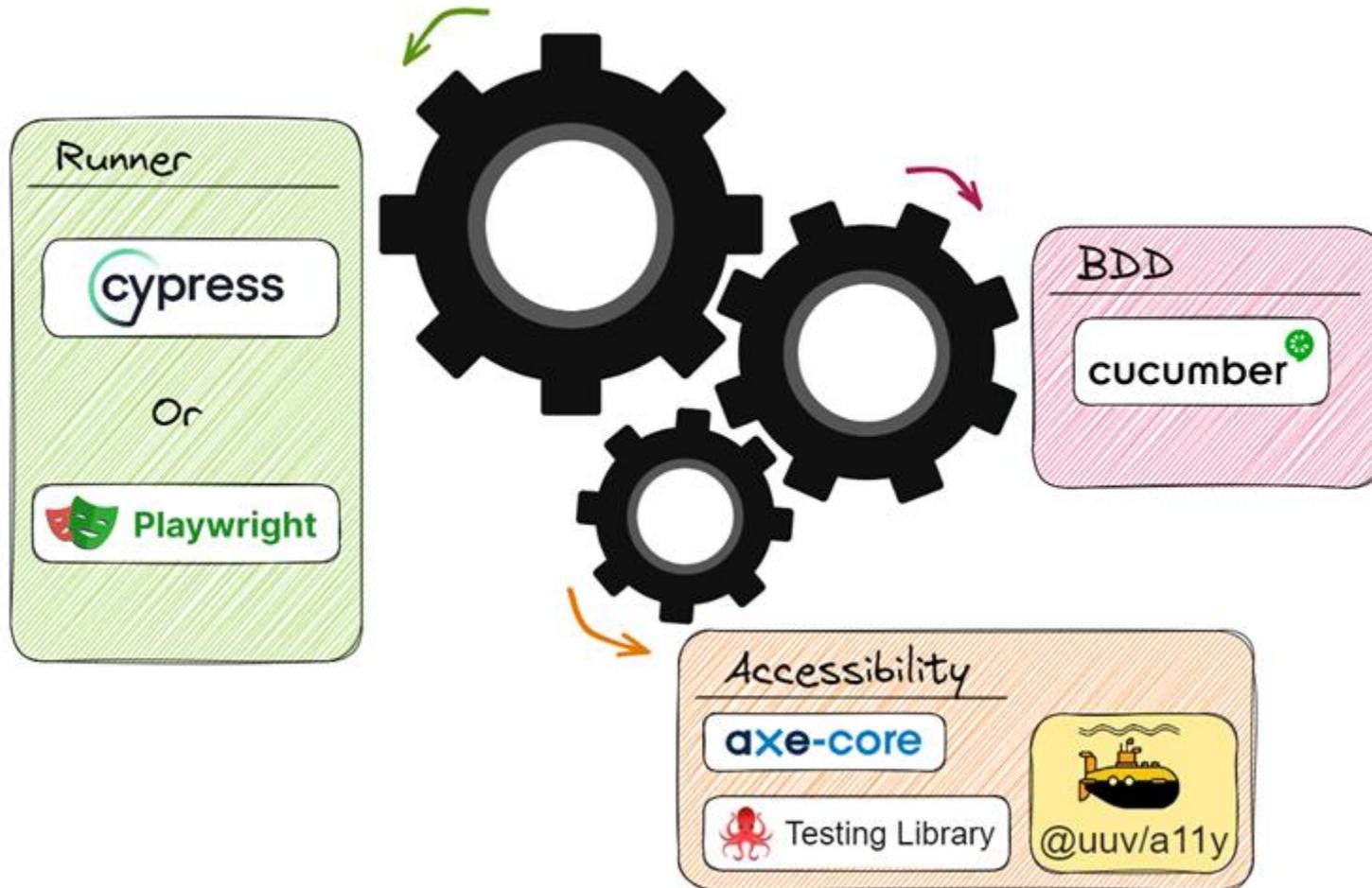


Solution basée sur l'**accessibilité numérique** pour **simplifier**  
l'**écriture de tests bout en bout** et vérifier le **respect des**  
**bonnes pratiques d'accessibilité**



# Ecosystème UUV

UUV Ecosystem





# Accessibilité Web

Bouton html



Styles Calculés Accessibilité Mise en page >>

- ▶ Arborescence d'accessibilité
- ▶ Attributs ARIA
- ▼ Propriétés calculées
- ▼ Name: "Ajouter un téléphone"
  - aria-labelledby: Non défini
  - aria-label: Non défini
  - Source : label: Non défini
  - Contenu: "Ajouter un téléphone"
  - title: Non défini
- Rôle: button
- Saisie utilisateur non valide: false
- Sélectionnable: true
- ▶ Visionneuse de l'ordre des sources

## Attributs Importants

**Nom  
Accessible**

**Rôle  
Accessible**



# Approche User Centric

Pour l'exemple suivant  
:

```
<body> <h1>Result</h1>... </body>
```

Library	Syntax
Cypress	cy.get('h1').contains('Result')
Playwright	await expect(page.getByTitle('Result')).toHaveCount(1)
Testing Library	expect(screen.getByTitle(/Result/i)).toBeTruthy()
UUV	Then I should see a title named "Result" Ou Alors Je dois voir un titre nommé "Result"

Dictionnaire de phrases en [anglais](#) et [français](#)



## Mise en place simple

- [Documentation](#) complète en anglais & français
- Dépendances NPM :
  - [@uuv/cypress](#)
  - [@uuv/playwright](#)
  - [@uuv/assistant](#)
  - [@uuv/a11y](#)
- Plugin JetBrains



## Aide à l'écriture des scénarios

- Dictionnaire de phrases en [anglais](#) et [français](#) prêtes à l'emploi
- [Possibilité de rajouter ses propres phrases](#)
- Plugin JetBrains UUV
- [UUV Assistant](#)



## Exécution des tests

- En **local** pour les développeurs
- Au sein d'un **CI** : Gitlab / Github ([exemple](#)) / Jenkins
- Rapport d'exécution **HTML** ([exemple de rapport](#))



# Comment améliorer l'accessibilité avec UUV ?

1. Ecrire les scénarios en respectant les [recommandations](#)
1. Utiliser pour toutes les pages les [phrases de vérification Axe-Core](#)
1. ***[Version Beta en cours]*** Utiliser les phrases liées au [module a11y](#)



## Fonctionnalités

- Mise en place simple
- Aide à l'écriture des scénarios
- Exécution des tests
- Amélioration de l'accessibilité numérique



Soirée du  
test logiciel

Conférence

# #4

## Perspectives



# Collaboration

- Collaboration avec des acteurs du monde a11y :
  - Tanaguru
  - Opquast
  - RAAM (Luxembourg)
  - D'autres experts a11y
  
- Présentations :
  - [TOSIT](#) : 09/11/2023
  - [Stand OSXP](#) : 06 & 07 décembre 2023
  - [Nantes JS](#) : 18 janvier 2024
  - [Live Angular Dev France](#) : 12 mars 2024
  - [Meetup à Bordeaux](#) : 20 mars 2024
  - [Soirée du Test Logiciel Bordeaux](#) : 21 mars 2024
  - [OW2 Con'24 breakout session accessibility](#) : 12 Juin 2024
  - [DevQuest Niort](#) : 14 Juin 2024
  - [Lyon JS](#) : 10 Juillet 2024
  - [Riviera Dev](#) : 07 Juillet 2025



## Prochaines fonctionnalités

- Correction des [bugs](#)
- Finaliser les tests de [parcours au clavier](#)
- Compléter le runner [Flutter](#)
- Finalisation du module [@uuv/a11y](#)



## Ce qu'il faut retenir

- Si bien utilisé, améliore l'[accessibilité](#) dès le développement
- [Living documentation](#) : Langage unifié pour développeurs et non développeurs avec un dictionnaire riche de phrases prêtes à l'emploi
- [Assistant](#) et le [plugin JetBrains](#) pour simplifier la rédaction des scénarios
- Intègre plusieurs moteurs d'exécution au choix : [Cypress](#) ou [Playwright](#)
- [Rapport d'exécution](#) user friendly et uniformisé
- [@uuv/a11y](#) pour effectuer des contrôles d'accessibilité



## A vous de jouer

- Essayer la solution :
  - [Kata E2E UUV](#)
  - Ou sur votre projet directement
- Nous faire un feedback
  - Publique : [GitHub Issue](#)
- Contribuer
  - [Liste des issues](#)

Laissez-nous une **star**  
ça nous encourage 😊





# #5

## Questions / Réponses



# Merci



Architecte Logiciel

[Make It Soft](#)



fr-dice



luifr10

Architecte Logiciel

[NobTech Consulting](#)



stanley-servical



stanlee974





## Autres ressources

- Github : <https://github.com/Orange-OpenSource/uuv>
- Demo with cypress runner engine : <https://www.youtube.com/watch?v=aTahsfoU-5w>
- Synthetic article : <https://medium.com/@njakolouis/user-centric-e2e-testing-1f99bdd6105f>
- Documentation : <https://orange-opensource.github.io/uuv/>
- npm playwright : <https://www.npmjs.com/package/@uuv/playwright>
- npm cypress : <https://www.npmjs.com/package/@uuv/cypress>
- npm assistant : <https://www.npmjs.com/package/@uuv/assistant>
- uuv/assistant interactive demo : <https://uuv-assistant.vercel.app/>

3ème édition de la  
Soirée du  
Test Logiciel  
Bordeaux

---



3 avril 2025  
17h à 22h30

---



École de Turing,  
Bordeaux



Soirée du  
test logiciel

 Bordeaux

**Merci  
de votre écoute !**