



**G**ZOLTAR

**Charles-Gaël Djedai**  
**Quentin Hinsinger**  
**Brendan Masson**  
**Aurélien Sieurin**  
**Florian Vrignaud**

**5INFO**  
**24 / 01 / 2014**

## Plan de la présentation

- Naissance de *GZoltar*
- Présentation de *GZoltar*
- Utilisation dans *Eclipse*
- Démonstration de l'outil

## Naissance de *GZoltar*

- **2005 : Idée d'automatiser les processus de test et de débogage logiciel**

Établir un diagnostique pour chaque cas de test

Informations concernant la couverture du code

- **2010 : Besoin d'améliorer la visualisation des rapports de test**

Première version de *GZoltar*

## Présentation de *GZoltar*

- **Framework**

Automatise les tests et le débogage d'un logiciel

- **Autre possibilité d'utilisation**

Permet d'implémenter de nouvelles techniques pour détecter et localiser des erreurs de tests

- **Fonctionnement**

S'installe en tant que plugin au sein d'Eclipse

S'articule autour des tests JUnit

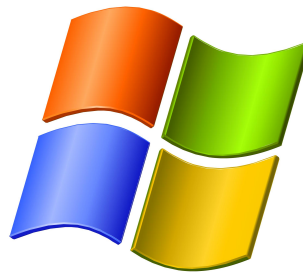
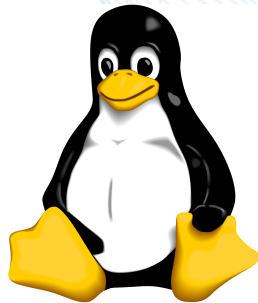
## Présentation de *GZoltar*

- Version actuelle : 4.0.1 (4 novembre 2013)

- En perpétuelle amélioration

Questionnaire d'évaluation lors du téléchargement de l'outil

- Framework multiplateforme



## Utilisation dans *Eclipse*

- Deux onglets dédiés à *GZoltar* dans *Eclipse*

### 1 - Regression-Zoltar

Lancement de tous les tests JUnit du projet  
Génération d'un rapport détaillant les erreurs

### 2 - Diagnostic Report

Visualisation des potentiels éléments fautifs  
Trois modes de visualisation

Exécution

Affichage

## Utilisation dans *Eclipse*

- Regression-Zoltar

Taille d'un cas de test

Détails des erreurs du test actuellement sélectionné

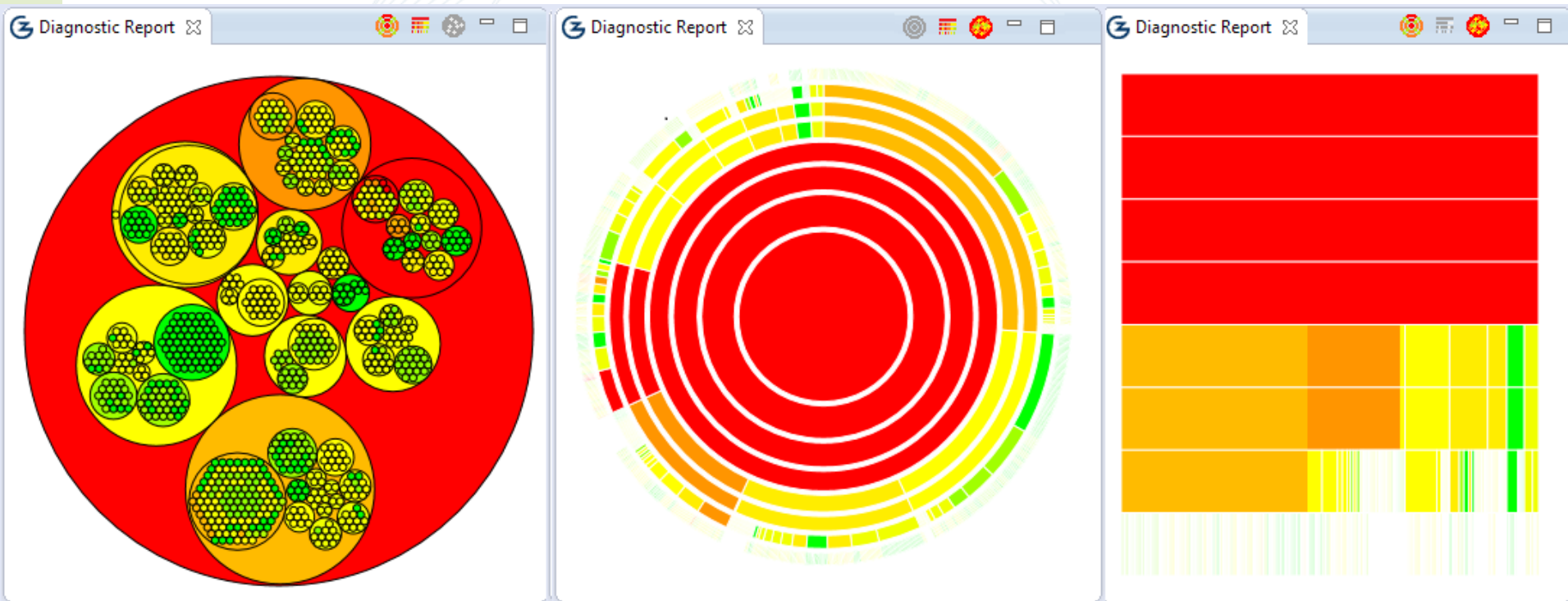
Set	Set Cardinality	Runtime (ms)	Failure Trace
Set 1	8	115	
nanoxml.test.java.InternalDTD		88	
nanoxml.test.java.Attributes		5	
nanoxml.test.java.Namespaces		5	
nanoxml.test.java.ComplexPe		3	
nanoxml.test.java.DoubleDTD		3	
nanoxml.test.java.Simple		4	
nanoxml.test.lite.Simple		5	
nanoxml.test.lite.Comments		2	
Set 2	8	118	
All Tests	9	121	junit.framework.AssertionFailedError at junit.framework.Assert.fail(Assert.java:48) at junit.framework.Assert.assertTrue(Assert.java:20) at junit.framework.Assert.assertTrue(Assert.java:27) at nanoxml.test.java.Attributes.testAttributes(Attributes.java:30) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(Unknown Source) at sun.reflect.DelegatingMethodAccessorImpl.invoke(Unknown Source) at java.lang.reflect.Method.invoke(Unknown Source) at junit.framework.TestCase.runTest(TestCase.java:168) at junit.framework.TestCase.runBare(TestCase.java:134)

Cas de test avec au moins une erreur

Temps d'exécution en millisecondes

## Utilisation dans *Eclipse*

- Diagnostic Report



*Bubble hierarchy*

*Sunburst*

*Vertical Partition*   
UNIVERSITÉ  
EUROPÉENNE  
DE BRETAGNE



## Utilisation dans *Eclipse*

- Diagnostic Report



L'élément possède une très faible probabilité d'être fautif



L'élément possède une probabilité moyenne d'être fautif



L'élément possède une forte probabilité d'être fautif



L'élément possède une très forte probabilité d'être fautif



# DÉMONSTRATION DE L'OUTIL

## Conclusion

- **Points négatifs**

Aucune communauté

En cours de développement

Peu de documentations (Présentation + Quick Manual)

API non documentée

- **Points positifs**

Facile à installer

Compréhension facile grâce aux couleurs