Talk
Tous les tests

Paul BEAUDET
Antoine DESRUET

Pourquoi tester?

```
entier fonction(entier a, entier b)
{
|    retour a < b;
}


entier mon_test(rien)
{
|    entier a = 1;
|    entier b = 1;
|
|    si (la_fonction_que_je_teste(a, b) renvoie vrai)
|    |    affiche("test OK");
|    sinon
|    |    affiche("test KO");
|    retour 0;
}
```
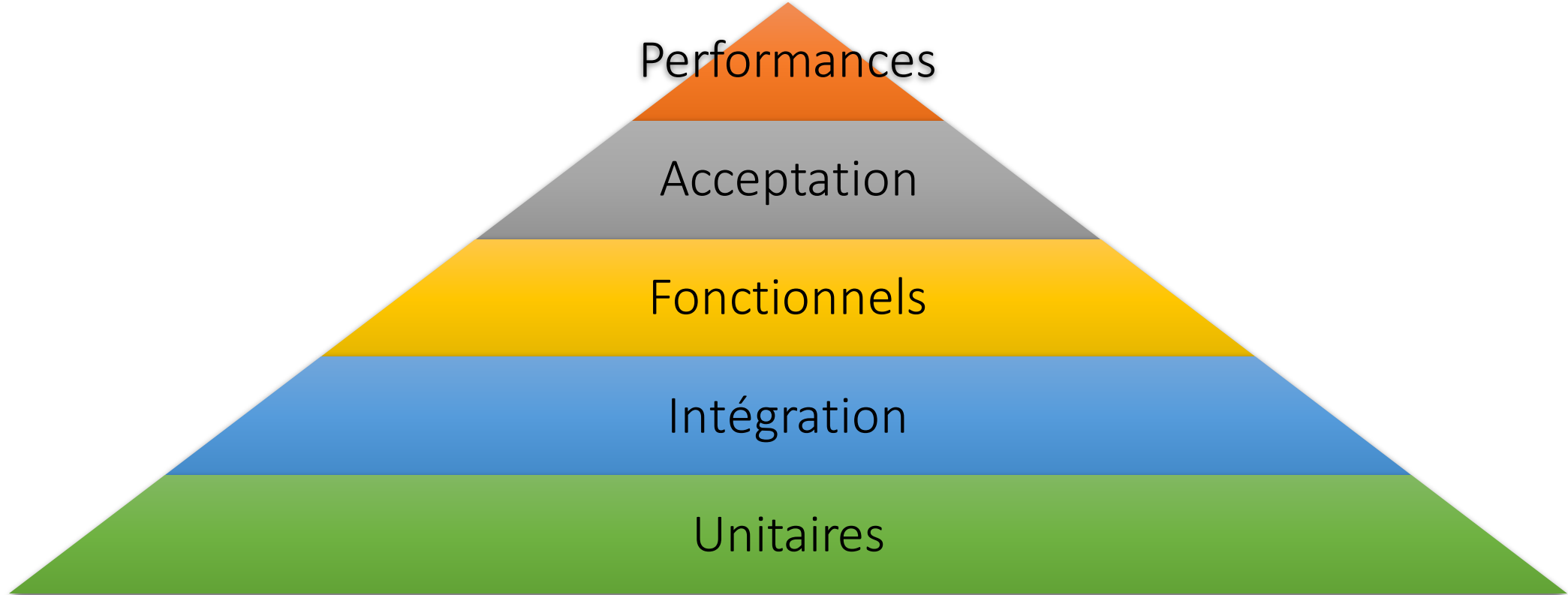
Différents types de tests

Tests unitaires

En C : Criterion


```
[====] Synthesis: Tested: 27 | Passing: 27 | Failing: 0 | Crashing: 0
-----------------------------------------------------------------------
                       GCC Code Coverage Report
Directory: .
-----------------------------------------------------------------------
File                              Lines    Exec  Cover  Missing
-----------------------------------------------------------------------
my_revstr.c                          11      11   100%
my_str_isalpha.c                      5       5   100%
my_str_islower.c                      5       5   100%
my_str_isnum.c                        5       5   100%
my_str_isprintable.c                  5       5   100%
my_str_isupper.c                      5       5   100%
my_strcapitalize.c                    9       9   100%
my_strcmp.c                           4       4   100%
my_strcpy.c                           6       6   100%
my_strlowcase.c                       5       5   100%
my_strncmp.c                          4       4   100%
my_strncpy.c                          6       6   100%
my_strstr.c                          10      10   100%
my_strupcase.c                        5       5   100%
-----------------------------------------------------------------------
TOTAL                                85      85   100%
-----------------------------------------------------------------------
```

En python: unittest

```python
import unittest

class Tests_my_min(unittest.TestCase):
    def test_ok(self):
        self.assertEqual(my_min(1, 2), True)
```

```
~/S/talk_tests    python3 -m unittest python_test.Tests_my_min
..
----------------------------------------------------------------------
Ran 2 tests in 0.000s

OK
```

```python
    def test_ko(self):
        self.assertEqual(my_min(1, 2), False)


if __name__ == '__main__':
    unittest.main()
```

{ Lab Astek }

En Haskell : HUnit

En Java : Junit et Google Truth

```java
import static org.junit.jupiter.api.Assertions.assertEquals;

@Test
public void whenComparingUsers_thenEqual() {
    User aUser = new User("John Doe");
    User anotherUser = new User("John Doe");

    assertThat(aUser).isEqualTo(anotherUser);
}

}
```

Tests fonctionnels

Tests d'intégration

```csharp
public class BasicTests
    : IClassFixture<WebApplicationFactory<RazorPagesProject.Startup>>
{
    private readonly WebApplicationFactory<RazorPagesProject.Startup> _factory;

    public BasicTests(WebApplicationFactory<RazorPagesProject.Startup> factory)
    {
        _factory = factory;
    }

    [Theory]
    [InlineData("/")]
    [InlineData("/Index")]
    [InlineData("/About")]
    [InlineData("/Privacy")]
    [InlineData("/Contact")]
    public async Task Get_EndpointsReturnSuccessAndCorrectContentType(string url)
    {
        // Arrange
        var client = _factory.CreateClient();

        // Act
        var response = await client.GetAsync(url);

        // Assert
        response.EnsureSuccessStatusCode(); // Status Code 200-299
        Assert.Equal("text/html; charset=utf-8",
            response.Content.Headers.ContentType.ToString());
    }
}
```
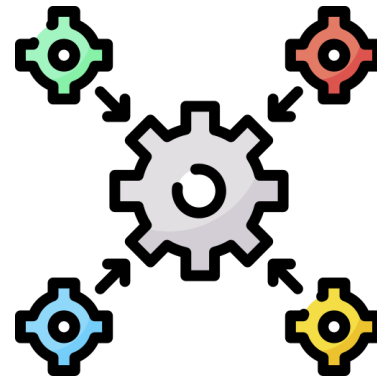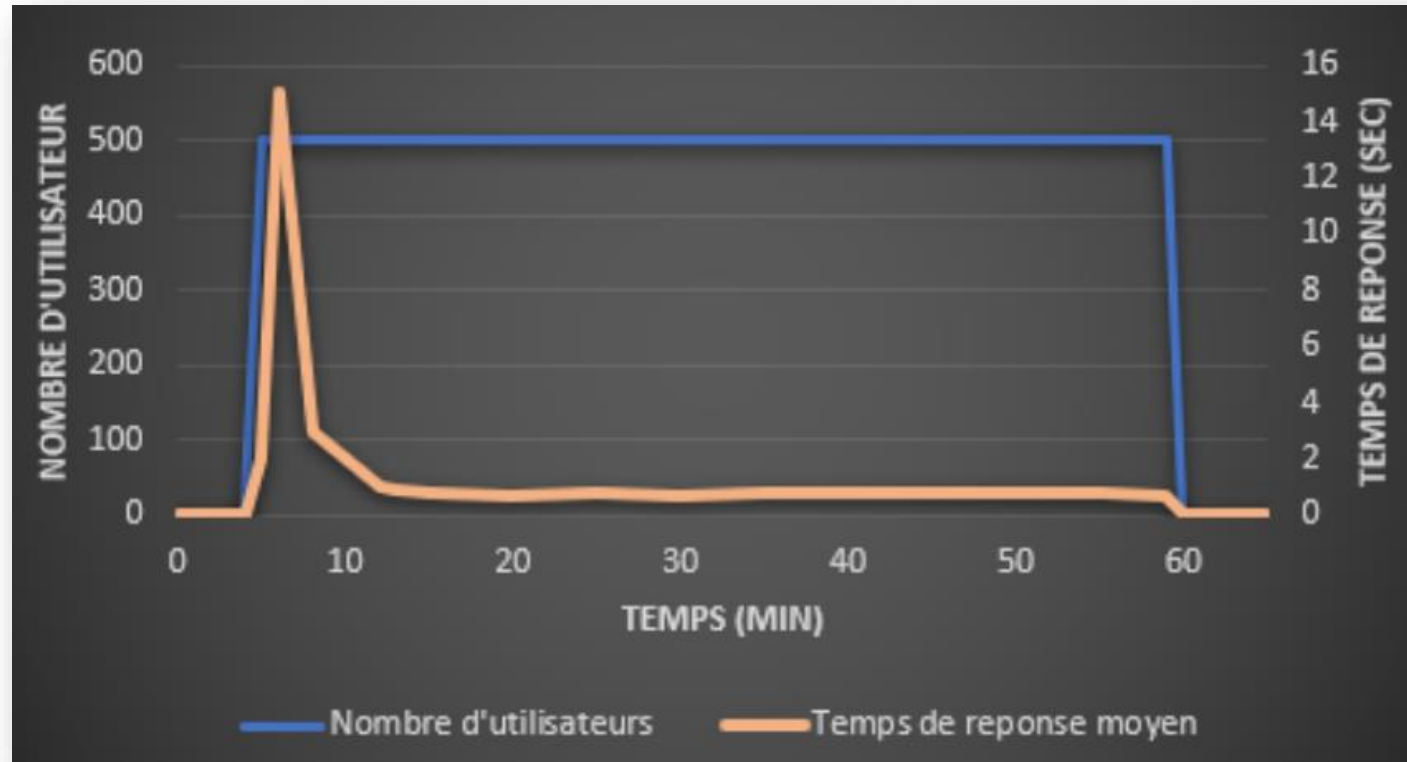
{ Lab Astek }

Tests d'acceptation

- Tests fonctionnels
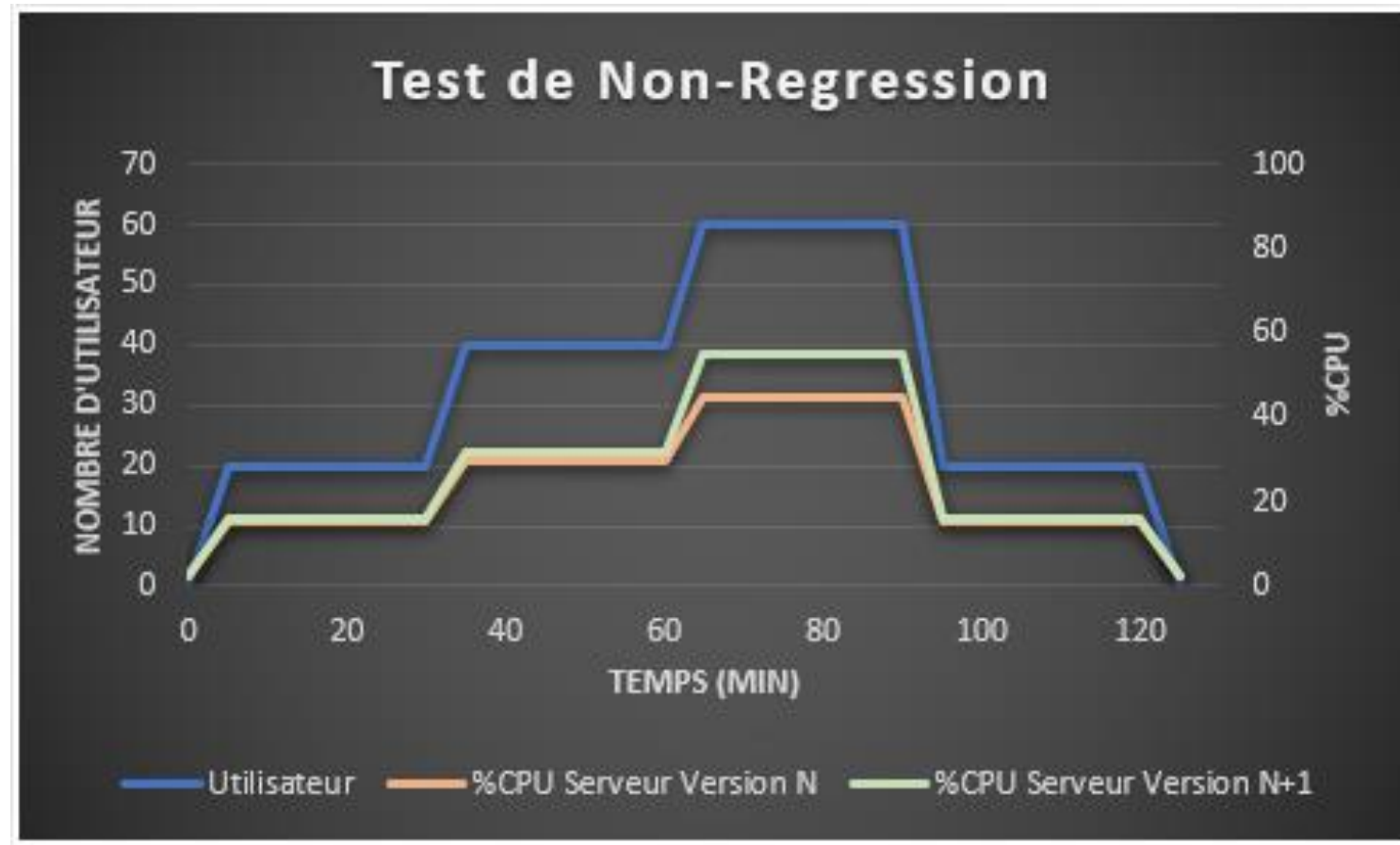- Tests d'intégration
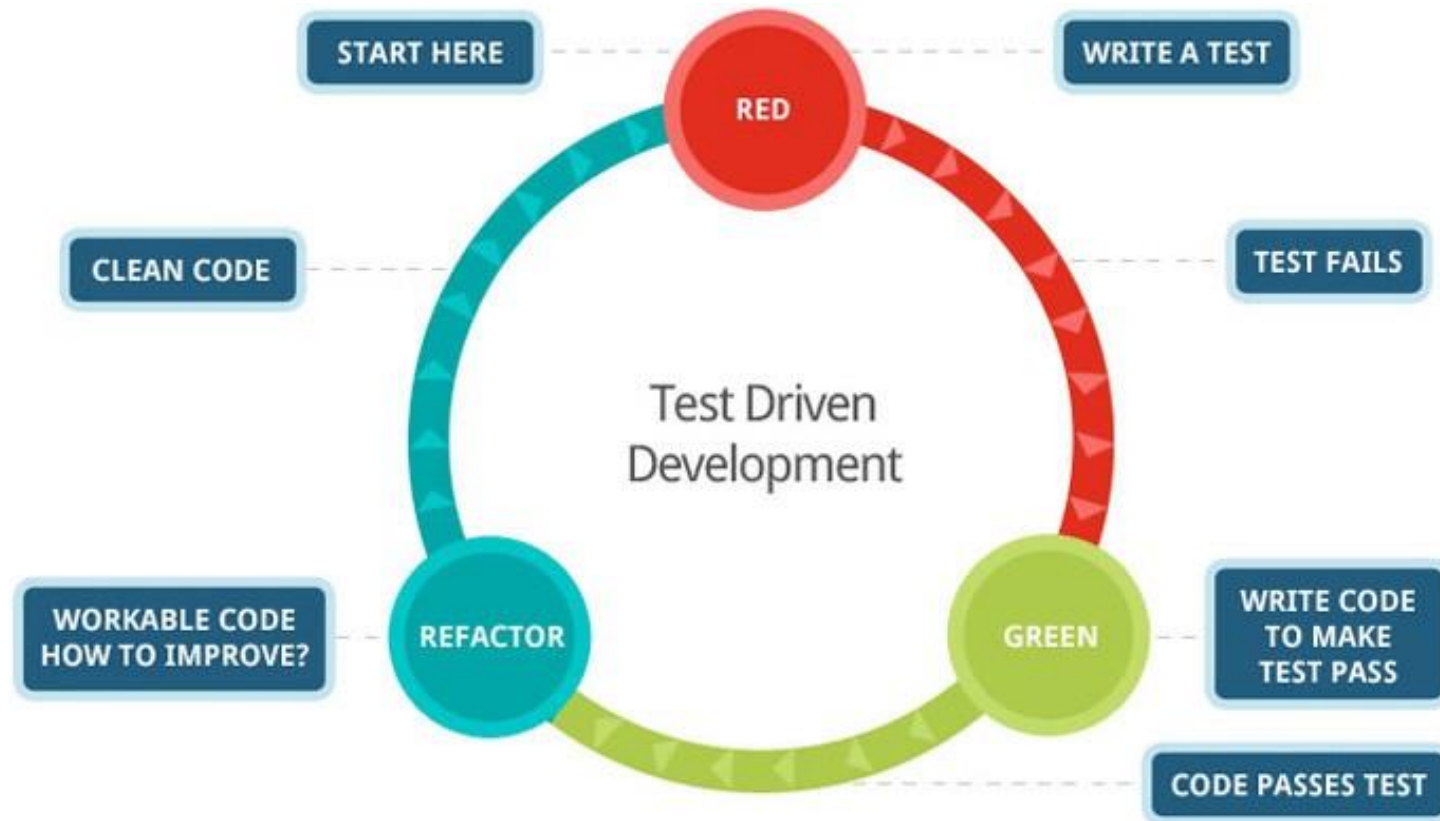- Tests de validation

Tests de performances

Tests de régression

Pour aller plus loin

{ Lab Astek }