

# VBScriptで ユニットテストをやってみる

UnitTestFrameworkの作り方

## アジェンダ

- 問題点の考察
- 作ってみる
- 使ってみる

# 問題点 その1

- そもそもビジネスロジックとHTMLが分離されていない

```
<%@ LANGUAGE="vbscript" CODEPAGE=65001 %>
<% Option Explicit %>
<html>
  <head></head>
  <body>
<%
  Dim isError
  . . . . .
  If isError Then
  > Response.write "<pre>" & vbCrLf
  > Response.write strErrorSql & vbCrLf
  > Response.write "</pre></BODY></HTML>"
  > Response.end
  End If
%>
  > <div class="calposit1">');"></div>
  <%If blnNextDay Then %>
  > <div class="calposit3"></div>
  > <div style="position: absolute; top: 13px; left: 885px; width: 810px;" class="button" onClick="onClick_Nex
<%
  > For i = 0 To nCnt
  > . . . . .
  > Next
%>
  > </div>
  <% Else %>
  > <% If strNextDate <> "" Then %>
  > <div class="calposit3">');"></div>
  > <%End If%>
  <%End If%>
  > <div id="caldate" class="calposit2">
  > > <span class="calcenter" onClick="onClick_Calender();"><%=strCalDate%></span>
  > </div>
  </body>
</html>
```



# 問題点 その2

- ユニットテストフレームワークが無い



# というわけで

- 無いなら作ればいい

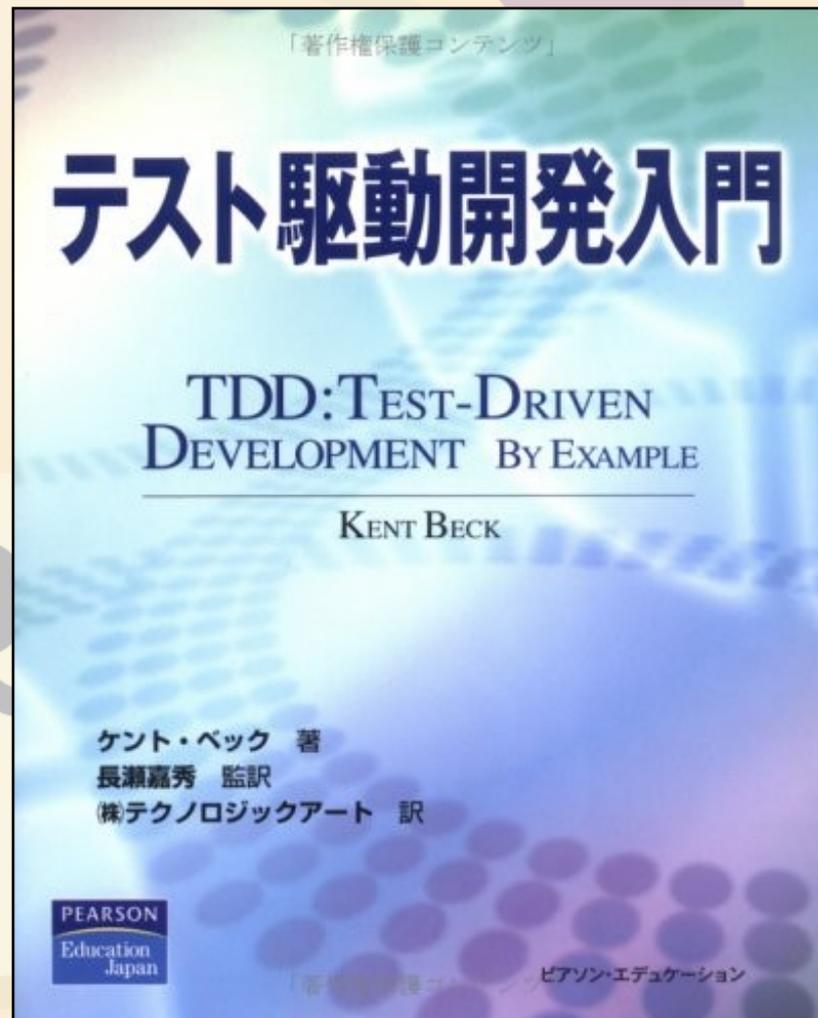
## テスト駆動開発入門

ケント ベック (著),  
長瀬 嘉秀 (翻訳),  
テクノロジックアート (翻訳)

出版社: ピアソンエデュケーション

ISBN-10: 4894717115

ISBN-13: 978-4894717114



わんくま同盟 名古屋勉強会 #08

# 作ってみる その1

- Phase 1

- とりあえず形を作る

- 実行用aspファイル

- vbsunit.asp

- ユニットテストフレームワーク

- vbsunit.inc

- テストコード

- testvbsunit.inc



# 作ってみる その2

- Phase 2
  - テストメソッドを呼び出す

```
testvbsunit.inc
見出し無し)
0 8 16 24 32 40 48 56 64 72 80
1 *%↓
2 Class TestCaseTest↓
3 > Public Sub TestTemplateMethod()↓
4 > End Sub↓
5 End Class↓
6 ↓
7 Sub main()↓
8 > Dim test↓
9 > Set test = New TestCaseTest↓
10 > test.Run()↓
11 End Sub↓
12 %>↓
13 [EOF]
```

# 作ってみる その3

- Phase 3
  - テストを実行する

```
testvbsunit.inc
Sub main()
15 > c_base.SetTest(sTest)↓
16 > End Sub↓
17 ↓
18 > Public Sub TestMethod()↓
19 > > s_log = "TestMethod"↓
20 > End Sub↓
21 End Class↓
22 ↓
23 Sub main()↓
24 > Dim test↓
25 > Set test = New WasRun↓
26 > test.SetTest("TestMethod")↓
27 > test.Run()↓
28 ↓
29 > If test.s_log = "TestMethod" Then↓
30 > > Response.Write("Success")↓
31 > Else↓
32 > > Response.Write("Failed")↓
33 > End If↓
34 End Sub↓
35 %>↓
36 [EOF]
```

# 作ってみる その4

- Phase 4
  - テストを数える

```
testvbsunit.inc
Sub main()
2  Class WasRun↓
3  . . . . .↓
4  End Class↓
5  ↓
6  Class TestCaseTest↓
7  . . . . .↓
8  > Public Sub TestTemplateMethod()↓
9  . . . . .↓
10 > End Sub↓
11 End Class↓
12 ↓
13 Sub main()↓
14 > Dim result↓
15 > Set result = New TestResult↓
16 ↓
17 > Dim test↓
18 > Set test = New TestCaseTest↓
19 > test.SetTest("TestTemplateMethod")↓
20 > test.Run(result)↓
21 ↓
22 > Response.Write(result.resultText)↓
23 End Sub↓
24 %>↓
25 [EOF]
```

# 作ってみる その5

- Phase 5
  - エラー処理の実装

```
testvbsunit.inc
Sub main()
2 Class WasRun↓
3   . . . . .↓
4   > Public Sub TestErrorMethod()↓
5     >   s_log = s_log & "TestErrorMethod "↓
6     >   Err.Raise vbObjectError + 1↓
7   > End Sub↓
8 End Class↓
9 ↓
10 Class TestCaseTest↓
11   . . . . .↓
12   > Public Sub TestErrorMethod()↓
13     . . . . .↓
14     >   test.SetTest("TestErrorMethod")↓
15     >   test.Run(result)↓
16 ↓
17     >   If test.s_log = "SetUp TestErrorMethod TearDown " Then↓
18     . . . . .↓
19   > End Sub↓
20 End Class↓
21 ↓
22 Sub main()↓
23   . . . . .↓
24   > Set test = New TestCaseTest↓
25   > test.SetTest("TestTemplateMethod")↓
26   > test.SetTest("TestErrorMethod")↓
27   > test.Run(result)↓
28 ↓
```



# 作ってみる その6

- Phase 6
  - AssertEqualsの実装

```
testvbsunit.inc
Sub main()
1 <%↓
2 Class WasRun↓
3   . . . . .↓
4   > Public Sub TestFailureMethod()↓
5     > Call assertion.AssertEquals("A", "B")↓
6   > End Sub↓
7 End Class↓
8 ↓
9 Class TestCaseTest↓
10  . . . . .↓
11  > Public Sub TestAssertion()↓
12  . . . . .↓
13  >   test.SetTest("TestFailureMethod")↓
14  . . . . .↓
15  >   If assertion.assertText = "required <A> but was <B>" Then↓
16  . . . . .↓
17  > End Sub↓
18 End Class↓
19 ↓
20 Sub main()↓
21  . . . . .↓
22  > Set test = New TestCaseTest↓
23  > test.SetTest("TestTemplateMethod")↓
24  > test.SetTest("TestErrorMethod")↓
25  > test.SetTest("TestAssertion")↓
26  > test.Run(result)↓
27 ↓
```

# 作ってみる その7

- Phase 7
  - 実行結果を表示

っていうか、成功・失敗は色で区別しないとね

成功: **Success**

失敗: **Failed**

エラー: **Error**



# 作ってみる その8

- Phase 8  
– 完成？



# 使ってみる

- サンプルプログラム