Экзамен по дисциплине: Тестирование ПО

Группа: СПУ 14в-1

Дата: 21.04.2016

Студент: Бутенко С А

1. Что такое дымное тестирование

Ответ:

Дымное тестирование проводиться после некого обновления продукты или сборки моделей продукта в единую систему. Дымовое тестирование проводиться поверхностно по основному функционалу продукта в поисках блокирующих и критических багов если они не были обнаружены это означает что дымное тестирование прошло успешную проверку.

1. Нефункциональное тестирование

Ответ:

Нефункциональное тестирование разделяется на:

Производительности

Установки

Удобства пользования

Безопасности

Отказ и восстановления

Конфигурационное

1. Регрессионное тестирование

Проверяет изменения продукта после исправления бага, слияния кода, перенос на другую ОС

Разделяется на три вида:

Регрессия бага- доказать что баг не исправлен

Регрессия старых багов-доказать что после исправление появились старые баги

Других дефектов -после исправление неисправна работает другая часть кода

Проверить систему аутентификации сайта http://mystat.itstep.org. Использовать WebDriver

Вариант правильный:

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Chrome;

using System.Threading;

using OpenQA.Selenium.Support.UI;

namespace UnitTestProject2

{

[TestClass]

public class UnitTest2

{

private string baseURL = "http://mystat.itstep.org";

private IWebDriver driver;

[TestMethod]

public void TestFF()

{

driver = new FirefoxDriver();

driver.Url = baseURL + "/";

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

WebDriverWait wait = new WebDriverWait(driver, TimeSpan.FromMinutes(5));

wait.Until(e => e.FindElement(By.Id("loginform-username")));

IWebElement login = driver.FindElement(By.Id("loginform-username"));

login.Clear();

login.SendKeys("Bute\_ap99");

IWebElement pass = driver.FindElement(By.Id("loginform-password"));

pass.Clear();

pass.SendKeys("\*\*\*");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

Thread.Sleep(3000);

Assert.AreEqual("Бутенко Сергей".ToLower(), driver.FindElement(By.XPath("//div[2]/div/div[2]/a/div")).Text.ToLower());

driver.FindElement(By.XPath("//a[3]/span")).Click();

}

[TestMethod]

public void TestChrome2()

{

driver = new ChromeDriver();

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).Clear();

Thread.Sleep(3000);

driver.FindElement(By.Id("loginform-username")).SendKeys("Bute\_ap99");

driver.FindElement(By.Id("loginform-password")).Clear();

driver.FindElement(By.Id("loginform-password")).SendKeys("qweasd123");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

Thread.Sleep(3000);

Assert.AreEqual("Бутенко Сергей".ToLower(), driver.FindElement(By.XPath("//div[2]/div/div[2]/a/div")).Text.ToLower());

driver.FindElement(By.XPath("//a[3]/span")).Click();

}

[TestCleanup]

public void ClinUp()

{

driver.Quit();

}

}

}

Вариант не правильный

using System;

using unit=Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading;

using NUnit.Framework;

using OpenQA.Selenium;

using OpenQA.Selenium.Firefox;

using OpenQA.Selenium.Support.UI;

namespace SeleniumTests

{

[TestFixture]

public class Untitled2

{

private IWebDriver driver;

private StringBuilder verificationErrors;

private string baseURL;

private bool acceptNextAlert = true;

[SetUp]

public void SetupTest()

{

driver = new FirefoxDriver();

baseURL = "http://mystat.itstep.org./";

verificationErrors = new StringBuilder();

}

[TearDown]

public void TeardownTest()

{

try

{

driver.Quit();

}

catch (Exception)

{

// Ignore errors if unable to close the browser

}

Assert.AreEqual("", verificationErrors.ToString());

}

[Test]

public void TheUntitled2Test()

{

driver.Navigate().GoToUrl(baseURL + "/");

new SelectElement(driver.FindElement(By.Id("city"))).SelectByText("Днепропетровск");

driver.FindElement(By.Id("loginform-username")).Clear();

driver.FindElement(By.Id("loginform-username")).SendKeys("Bute\_ap99");

driver.FindElement(By.Id("loginform-password")).Clear();

driver.FindElement(By.Id("loginform-password")).SendKeys("\*\*\*");

driver.FindElement(By.XPath("//button[@type='submit']")).Click();

try

{

Assert.AreEqual("× Неверный логин и/или пароль", driver.FindElement(By.Id("w0")).Text);

}

catch (Exception e)

{

verificationErrors.Append(e.Message);

}

}

private bool IsElementPresent(By by)

{

try

{

driver.FindElement(by);

return true;

}

catch (NoSuchElementException)

{

return false;

}

}

private bool IsAlertPresent()

{

try

{

driver.SwitchTo().Alert();

return true;

}

catch (NoAlertPresentException)

{

return false;

}

}

private string CloseAlertAndGetItsText()

{

try

{

IAlert alert = driver.SwitchTo().Alert();

string alertText = alert.Text;

if (acceptNextAlert)

{

alert.Accept();

}

else {

alert.Dismiss();

}

return alertText;

}

finally

{

acceptNextAlert = true;

}

}

}

}