# *Systems III (420-E31-HR)*

# *Lab 08 – Defect Management*

Date assigned: Wednesday, October 25, 2017

Date due: **Wednesday, October 25, 2017, 12:00 p.m.**

**Learning Objectives**

Upon successful completion of this lab exercise, the student will be able to:

* Understand severity and priority in defects
* Produce guidelines for creating a well written bug report
* Create a defect in TFS

To do:

Create a new folder named **YourUserName\_E31\_L08\_Defects** in your 420-E31 folderin your home drive. Save this document as a Word document named **YourUserName\_E31\_L08\_Defects.docx** in the folder. The document will hold your answers for your lab

**Part A – How to Write a Good Bug Report**

When defects are written poorly, it can reduce the productivity of the team. A poorly written bug report also reduces the chances of the bug being reproduced and fixed.

Prepare a **one** page document on how to write a good bug report. It needs to be comprehensive, yet to the point. Focus on the title, description, steps to reproduce the problem and expected/actual results. **(15 marks)**

The following sites are good references:

<http://university.utest.com/writing-quality-bug-reports-and-utest-etiquette/>

<http://noverse.com/blog/2012/06/how-to-write-a-good-bug-report/>

**Part B – Bug Classification**

1. Based on the following scales what severity and priority would you attach to the following bugs? Provide a justification for your decision in each case. **(11 marks)**

|  |  |
| --- | --- |
| Severity | Priority |
| 1 – Crash | 1 – Critical: Immediate Fix Required |
| 2 – Subsystem not working | 2 – High: Must fix before Release |
| 3 – Feature not working | 3 – Medium: Fix before release if time |
| 4 – Minor problem, Known workaround, spelling mistake, etc | 4 – Low: Post-release fix |
| 5 – Suggestion | 5 – Not required for this release |

* 1. The system crashes when an invalid password is entered.

Crash – crucial

Login in is potentially necessary to do anything with the system

* 1. Data can be entered, but no reports can be generated.

Feature not working – medium

We don’t know how important this is for the system, but it’s a feature that could come in a later release.

* 1. The company logo is displayed inverted on the Splash screen.

Suggestion – critical

Bad image to the company, you want people to recognize it

* 1. The system cannot print to PostScript printers, but can to all other types of printers. The company executives rely solely on PostScript printers.

Feature not working – high

They’re the most important users of the system and they can’t use it.

* 1. The system crashes every time the date changes to February 29 unless a restart is performed sometime in the week preceding February 29.

Minor problem, known work around – Low priority or not required for release

This problem has a quick and easy workaround, a fix can come in a later release.

* 1. The system crashes every time the year changes.

Crash – low or not required for release

Crashes only once a year, fix can come in a later release

* 1. When entering a new customer, the system fails to produce an error if a mandatory field is missing. The system does not progress to the next screen, but does not tell the user what the problem is.

Subsystem not working – high

Leaves a user totally clueless. Don’t know what the system is and therefore how often this will happen, but it’s important.

* 1. When entering a new customer, the system fails to produce an error if a mandatory field is missing. The system does not progress to the next screen, and the customer data already entered is not maintained (that is, it must all be re-entered).

Subsystem not working – high

No error produced confusing the user and all data entered deleted, frustrating the user.

* 1. The help file contains numerous spelling errors.

Spelling mistake – medium

Doesn’t effect system usability but could effect company image.

* 1. The French version of the online tutorial does not match the English version.

Minor problem – medium

As long as it gets a similar message across, it’s not high priority.

* 1. The installation program fails when there is insufficient disk space and does not produce an error message.

Suggestion – low

Installation is expected to fail, but it should give an error message.

* 1. The installation program fails when there is insufficient disk space and produces an error explaining that there was not enough disk space.

Minor problem – not required

I see little problem with this. If there’s no space it’s expected to fail, error message is good.

**Part C– Assignment 3 Preparation (15 marks)**

1. Read Assignment 3. Provide a 2 paragraph description of the tasks needed to be done. **[5 marks]**

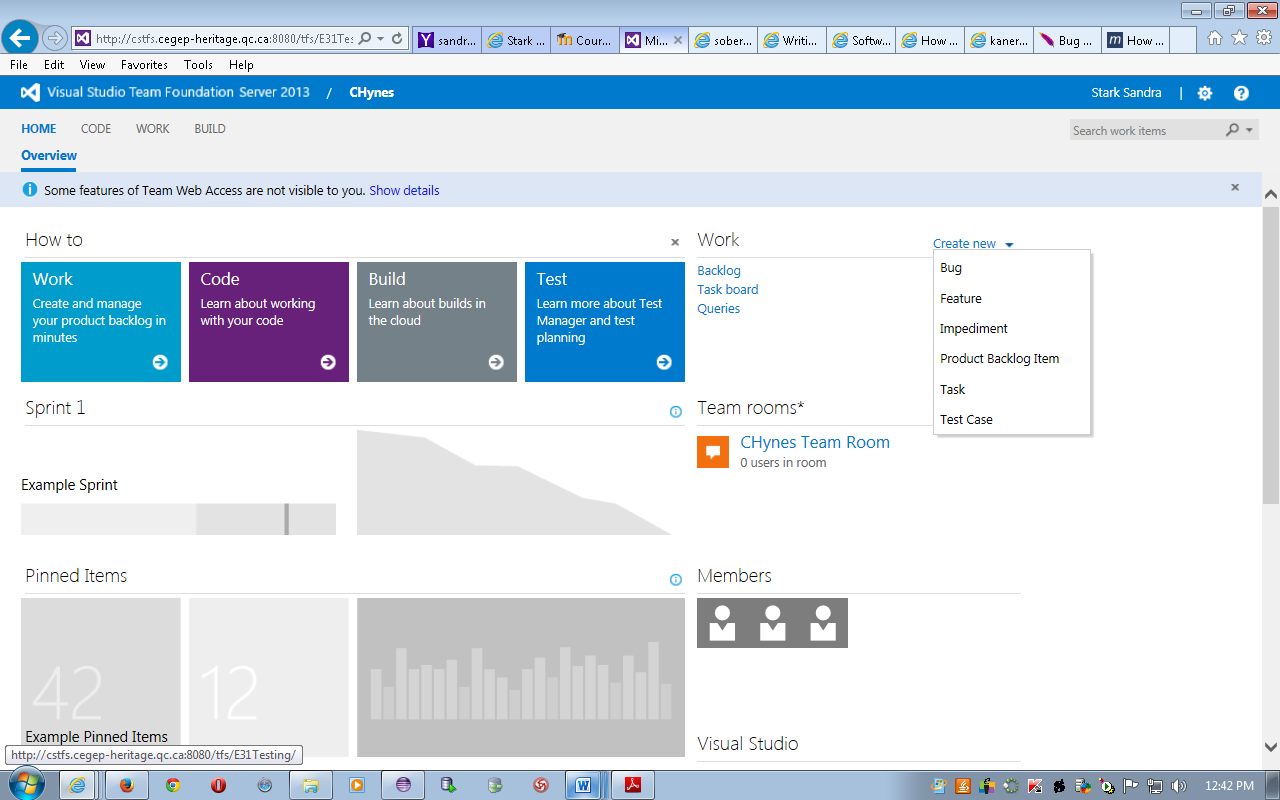
In Part A of the assignment we need to design test case matrixes for the course evaluation system. The test cases need to cover everything out of the schedule evaluation use case narrative on moodle. After creating a test case matrix with simple valid or invalid in the boxes, you need to go and repeat the test matrix replacing simply valid or invalid with actual values that satisfy the condition.

Part B of the assignment is much more complicated than part A. For part B we need to write a test case plan and perform multiple different types of testing on our own test environments for the course evaluation system. We need to perform usability tests, security tests, compatibility tests, etc., across the entire system.

1. Complete Part A. Show me when complete for marks. Keep it saved in your assignment folder. **[5 marks] banana2**
2. Jump to Part B #9. Familiarize yourself with TFS application located at <http://cstfs.cegep-heritage.qc.ca:8080/tfs>

Find 2 defects with the CES system and log them in TFS. Instructions below. **[5 marks]**

* 1. On the TFS welcome page, select **Browse**, under the Recent Project & teams section. You should see an **E31Testing** project listed.
  2. Select the **E31Testing** project. At this point you should see your userid listed under the **E31Testing** project. Select your user id and click the **Navigate** button.
  3. This should take you to the main TFS page. From here, select **Create New** and then **Bug** from the dropdown list.



* 1. Add a descriptive title for your defect.
  2. Under the **Assigned To** drop down list, select **Fox Christopher**.
  3. Assign an appropriate severity and priority to the defect.
  4. Under the **Steps to Reproduce**, enter the comprehensive list of steps to reproduce the problem.
  5. Under the **Acceptance Criteria** tab, enter the criteria for this bug to pass in order to close the bug as fixed.
  6. Notice that there is a tab for **Test Cases**, where you can enter the test case that found this bug.
  7. Notice that there is an **Attachments** tab, which is useful to include screenshots from the testing.
  8. Save the bug and return to the main TFS page.
  9. Navigate to the **Work, Backlog** page. From here, you should see the bug that you just raised. If you don’t see it listed, something is wrong…

**To submit**

When you have completed the assignment, upload **YourUserName\_E31\_L08\_Defects**, to the Moodle page for this course.