# *Development Project II (420-E63-HR)*

# *Assignment 1 – Setup - Review, reassess and refactor*

Date assigned: Friday, January 19, 2017

Date due: **Friday, January 19, 2017, 3:50PM**

**Computer Late assignment policy does not apply. It’s due at the specified due date.**

**Learning Objectives**

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

* Review the process standards defined in Development Project I.
* Add bug descriptions to the project issue tracker
* Add Refactoring tasks to the project issue tracker
* Prioritize, as a team, the outstanding bugs and refactoring tasks
* Analyze results from Development Project I and propose changes for improvements.

To do:

This assignment is to be completed as a team and individual activity. For the individual part of the assignment, create a new Word document named **YourUserName\_Teamxx\_E63\_A01\_SetupAndPrep.docx.** Include your (short) username and team number.

# Section 1 – Individual Work

## Part A – Outlook Setup

1. Set up the blocks in your timetable in the calendar in Outlook using recurring appointments, with the *Range of recurrence* to *End by*: May 11, 2018, which is the last day of classes. Change any exceptional dates, such as the study week, midterm holiday, the Easter holidays and any days that follow a different schedule, such as a Monday that follows a Thursday schedule, which can be found in the Academic Calendar link at the top of the college web site.
2. Share the calendar with the Computer Science teachers.

Complete the following checklist:

|  |  |
| --- | --- |
| Calendar updated with regular schedule of classes/labs/etc. |  |
| Updated with exceptional dates/changes |  |
| Calendar shared with CS teachers |  |

## Part B – Organization and Plan

*Individual Work*

Analyze this assignment and plan.

1. Identify the individual deliverables:

The individual deliverables for this assignment include:

* the updated outlook calendar
* A list of individual and team deliverables
* Adding bugs to TFS
* Add refactors to TFS
* Review the code and identify standards violations

1. Identify the team deliverables:

The team deliverables for this assignment include:

* We need to prioritize and size our bugs that were listed in the system
* We need to do the same for refactoring of our code
* We need to provide a memo for the meeting and minutes.

1. What is your estimate on when the individual portion will be complete? Negotiate with your team to agree upon a time you’ll meet and organize to start working on the team portion. State your estimate and the team rendezvous time as well as any concerns or risks on hitting this.

* I estimate that I will be done the individual portion around 10:30. I got sidetracked figuring out taking Calculus, so it’ll be a bit longer.

## Part C – Updating Work items

*Individual Work:*

Based on your Final comprehensive assignment of 420-E50-HR

1. Add your System Test bugs to TFS. Your bug report must have the details required to reproduce the bugs. Be sure to classify this as a “bug” work item and not a product backlog item.
2. Add Refactoring Proposals as a backlog item with tag “refactor”.
3. Do not worry if there are duplicates with someone else’s bug. You’ll clean that up later.

**Update the tables below with your updates:**

**Bug List**

|  |  |
| --- | --- |
| **TFS ID/URL** | **Bug Title** |
| 178 | Captcha isn't registered for CSDEV |
| 179 | User can see program competencies when they selected only genEds |
| 180 | User can't remove files |
| 183 | The candidate can see too many commons that they shouldn't have access to |
| 186 | The system doesn't log off a profile properly |
| 189 | The view RAC request page is missing data and some is inaccurate |

**Refactor Task List**

|  |  |
| --- | --- |
| **TFS ID/URL** | **Backlog Item** |
| 193 | Clean up cshtml files |
| 195 | Make JavaScript files more modular |
| 199 | Make CandidateBLL class more modular and readable. |

## Part D – Process and Standards

1. Review the team’s coding standards document and reconcile against samples of your code. We will be holding code inspections later on. Ideally, your team’s code meets this standard. Find at least 3 defects.

|  |  |
| --- | --- |
| **Defect parameter** | **Possible values** |
| **Origin** | **Requirements, Design, Implementation, Testing** |
| **Type** | **Missing, Wrong, Extra, Usability, Performance, Style, Clarity** |
| **Severity** | **Major, Minor** |

**Code inspection defect report**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Filename/line#** | **Origin** | **Type** | **Severity** | **Description/Comments** |
| **CandidataBLL, 170** |  | **Style** | **Minor** | **One line of code spans 218 characters, instead of wrapping at our 80-character limit.** |
| **CandidateBLL, 171-176** |  | **Extra, Clarity** | **Minor** | **These 6 lines of code can be replaced by a single Boolean comparison – so much redundant code.** |
| **CandidateBLL, everywhere** |  | **Style** | **Minor** | **There’s many places where there’s braces for conditional statements, loops, catch blocks, etc., where there’s a space padding the braces and the contents. This violates the coding standards.** |

1. Review the team’s coding standards document. If the document is wrong or needs to be updated, now is your chance to make some proposed changes.

Fill in the table below with your proposal. No more than 5 items.

**Proposed Changes to coding standard**

|  |  |  |
| --- | --- | --- |
| **Proposal #** | **Category** | **Proposal/Description** |
| **1** |  | I would like to increase the line limit to 100, not 80. It’s still very easy to see all in one line (If I half-screen VS, I can still see full 100 lines on default font-size) |
| **2** |  | I would like to decrease the line wrapping spaces required. Right now we line wrap with 8 spaces, but I’d like to keep it at 4 because otherwise we’ll end up with lines wrapping much more than needed. |

1. Reflect on last semester’s sprints.

Identify what new lessons learnt should be added.

What behaviours, approaches or processes would you change to improve the sprint process this semester to improve in the following categories:

* 1. Ability to deliver and complete functionality/PBI
  2. Ability to deliver quality (cover code standards, white and black box testing)
  3. Team and individual evaluation and scoring
  4. Meeting overhead (scrums, sprint reviews/planning, retrospectives).

**Proposed Changes to Process**

|  |  |
| --- | --- |
| **Category** | **Proposal/Description** |
| **Ability to complete PBI** | We should spend more time on proper acceptance testing, because a lot of the time we had things done, but not the way Alain quite wanted, so they ended up carrying over to the next sprint. |
| **Ability to Deliver quality** | We have very little code coverage for unit tests, which impact the quality of the software, so we need to improve our unit test coverage |
| **Team and Individual Scoring** | I see nothing to be changed with how the grading was done, either team or individual. |
| **Meeting Overhead** | We need to go to our meetings more organized, because we we’re often scrambling in the meeting to say things. We need to prepare for them and come to actually be prepared because they should go quickly if we prepare. |

# Section 2 – Team work

Only one person on each team needs to submit the team portion of this assignment. Nominate/action a team member to do this. Capture these items in the form of a single meeting minutes memo capturing what was discussed, decisions, and any new, open actions. Capture what actions were done by whom and all decisions.

Make sure the team reviews the minutes before they are submitted.

There will be a follow-up meeting later to discuss the team proposals with your project manager.

**Sync up with the team manager before any team deliverable is submitted.**

**All individual submissions should be complete before working on the team portion. If this is not possible or practical for your team, please discuss with the project manager.**

**See Moodle for details on how to submit the team portions. Discuss with the project manager if there are any clarifications needed on the deliverables or how to submit.**

1. Update the Action Item Register to cover all open action items from last semester. We will use this to track all actions and the daily scrum commitments. Get together as a team and review all past actions to get agreement on what was closed and what remains open.
2. Update the Lessons Learnt tab to cover all the lessons learnt from last semester.
3. Discuss any individual coding standard change proposals. Determine if there are any agreed upon changes and if so capture these in the updated standards document and in the minutes.

Once all the individuals have completed their bug list and Refactor Task List, get together and discuss

1. Prioritize and order the bugs and Refactoring items. For now, order this above the customer-visible backlog items. We have about a week to get this cleaned up before we resume sprints.
2. Roughly size the bugs and Refactoring items. (enough to understand what’s small/medium/big/enormous). (Re-prioritize and order if you need to).
3. Send an email to the team and the project manager stating your “Refactor & Bug fix plan” and commitment. This is just like a sprint goal/commitment email. Ensure that you:
   1. Provide a screen capture of your team’s agreed upon backlog list.
   2. Be clear which items the team is committing to. Use the iteration label Release1/Refactor1 which is scheduled as Friday Jan 19 to Thursday Jan 25.
   3. List the resource availability to do so. i.e. total number of person hours applied to an estimate of N hours.
   4. Do not send this to the end user. This is an internal matter.
4. Discuss your proposed changes to process and capture the agreed upon proposals:

What behaviours, approaches or processes would you change to improve the sprint process this semester to improve in the following categories:

* 1. Ability to deliver and complete functionality
  2. Ability to deliver quality (cover code standards, white and black box testing)
  3. Team and individual evaluation and scoring
  4. Meeting overhead (scrums, sprint reviews/planning, retrospectives).

As a team, collate, discuss and reach agreement on proposed changes and update the table below:

**Proposed Changes to Process (Team)**

|  |  |
| --- | --- |
| **Category** | **Proposal/Description** |
|  |  |
|  |  |

# Section 3

## Part A – Work on backlog prioritized list

*Individual Work:*

Take a task (bug or refactoring) from your prioritized list for the Refactor iteration and get working!

Do NOT work on any task that has not been committed to without agreement with your team and your project manager.

Follow the process. All changes must be reviewed.

There will be a team assessment done for this next week. You will scrum daily on this and treat it like a regular iteration. We’ll have a quick review on the iteration results later in the week.

**To submit**

When you have completed the assignment, upload the **YourUserName\_E63\_A01\_Setup.docx** document to Moodle.