# *Maintenance Project (420-E71-HR)*

# *Task 3 – Plan I*

Date assigned: Monday, February 5, 2018

Date due: **Monday, February 5, 2018 by 12:00**

**Learning Objectives**

Upon successful completion of this task, the student will have:

* Meet with the end user to gather requirements and priorities
* Analyzed, prioritized and cleaned up the backlog items
* Propose a project methodology to use for the project
* Set up the development environment for the project

To do:

1. Save this document with the name **YourUserName\_E71\_A03\_Plan.docx** in the 420E71 folder of your H drive. Submit a single copy of this as a team. Please put the name of the person responsible for each item, else, you will get no credit for work completed.

### Part A – Plan of the Day

**8:15 am kickoff meeting** – Make sure you’ve read this whole document. Organize as a team and present to me how all this will be done by noon today, who will do what tasks. There are 9 of you so organize well.

Capture your plan. List in sequence order. Not every item needs a person assigned as you will do FCFS after your initial task (add more rows as required).

Hint: Either capture this in a table below, or just whiteboard a Kanban and include before and after photos below:

|  |  |
| --- | --- |
| **Work item** | **Person** |
| Capture Plan of the Day |  |
|  |  |
|  |  |

### Part B – User meeting – Gather context, requirements and priorities

1. Generate Meeting Minutes of your meeting with the user (Allan) on Monday January 29, and send to all attendees. Be sure to capture who was present, what went on, key decisions and agreed upon next steps or action items. Make sure this is reviewed and agreed upon by the team and your professor. Embed the minutes in this document here:
2. Capture all user requests and all of your bugs found in TFS. Use the following use case template (see template + example on moodle).

|  |  |  |  |
| --- | --- | --- | --- |
| Story {TFS #}  **{name}** | | | |
| **Business Rule** | {#} | **Epic** | {name} |
| **Priority** | High Medium, Low} | **Size** | {small, medium, large} |
| **Story Narrative** | | **Acceptance Criteria 1** | |
| **As a** | {role} | **Given** | {context} |
| **I want** | {something} | **When** | {event 1} {event 2} |
| **So that** | {value} | **Then** | {outcome 1} {outcome 2} |
| **Comments:** |  | **Acceptance Criteria 2** | |
|  |  | **Given** | {context} |
|  |  | **When** | {event 1} {event 2} |
|  |  | **Then** | {outcome 1} {outcome 2} |
|  |  | **Acceptance Criteria 3** | |
|  |  | **Given** | {context} |
|  |  | **When** | {event 1} {event 2} |
|  |  | **Then** | {outcome 1} {outcome 2} |

### Part C – Backlog/Change Review

We are going to meet as a team to discuss:

* what work will be done
* what the timeframe for the work will be
* what role each team member will play

As a team, you will come up with a proposal to present to me (the PM) for approval.

1. Clean up TFS backlog. Organize on how to get rid of duplicate backlog items and stale backlog and execute on this plan. Your goal is to make the backlog list accurate and current.
2. As a team, agree on the severity and priority (order 1-highest, first to do) of the backlog list. The priority must be consistent with the user’s stated priorities.
3. Take some time to go over the architecture of the project, the solution’s code, and discuss as a team what will be needed to rewrite the application. Go through the different layers (presentation, business & data layers), the different components, and write a work item for each activity.
4. Decompose and size your top priority items on the backlog list. You only need to go as far as you think you could possibly attain during the first sprint/iteration. i.e. if there’s 10 items and you only think the first 5 are achievable, you don’t need to size the items you wouldn’t be able to address. Meet as a team to discuss what needs to be done. Come up with a consensus on steps, who is doing what, etc. Come up with work items that are needed and attach to defects and user stories as needed. Include time/effort estimates in the work items/defects/user stories. Use either planning poker or relative table top estimation for your sizing.
5. When you have determined the workload, meet with me. We will review and estimate the defects. Your proposal on the process, roles, timeframe and content.

### Part D – Set Up Your Development Environment

1. Setup your TFS to access the source code. Remember that you’ll be using the CSAdmin-2018-MR-Team1/Team2 branch of CSAdmin.
2. **Each individual must have his own database**. Remote login to csdev to create your own copy. Remember to change your connection string to point to your own DB copy and NEVER check in your connection string change into the maintenance release. Leave the maintenance release working from a common database.
3. Make sure that you can run your project and that you are pointing to the correct database (your own).
4. Baseline/compare your project to the version running on CSDEV that you used in your initial analysis. Note any and all differences you see (this should be compared to the group you are working with).
5. Fill in the table below (add additional rows as the team sees fit). Put a check mark in for each completed task per person:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *<Student name>* | *<Student name>* | *<Student name>* | *<Student name>* | *<Student name>* | *<Student name>* |
| Personal copy of source code CSAdmin |  |  |  |  |  |  |
| Personal copy of database with revision control (Redgate) |  |  |  |  |  |  |
| Verified personal environment, up and running |  |  |  |  |  |  |

I propose that a single team member scope this out first in case there are any issues rather than the whole team struggle with the same issues.

**To submit**

When you have completed the task:

* Save this file and submit to the appropriate location in Moodle.