**HWK 6**

**Snehadarshini ( Darshii )**

**Repository Link :** <https://github.com/snehadarshini-pfw/ACS_567_HWK/branches>

**Branches Link** :

* <https://github.com/snehadarshini-pfw/ACS_567_HWK/tree/HWK6-Task1>
* <https://github.com/snehadarshini-pfw/ACS_567_HWK/tree/HWK6-Task2>
* <https://github.com/snehadarshini-pfw/ACS_567_HWK/tree/HWK6-Task3>
* https://github.com/snehadarshini-pfw/ACS\_567\_HWK/tree/HWK6-Task4

**Feature Break down into Tasks.**

Task 1: Create a spring boot starter project

Time required: 30 minutes.

Task 2: Create a post endpoint for Feature A.

Time required: 30 minutes.

Task 3 : Create a post endpoint for Feature B

Time required: 120 minutes.

Task 4: Creation of documentation.

Time required: 60 minutes.

Sprint Duration: 2 Days

**Parallel work that can be done**

* Task 1 is high priority Task.
* Task 2 and Task 3 can be done in parallel after completion of Task 1. Since we are creating 2 different end points for Task2 and Task 3, they can be done in parallel. The logic that needs to be implemented for both is different and not dependent on each other. Hence these two tasks can be done in parallel but after the completion of Task 1. Since it is just 1 developer as of now, all the tasks will be done sequentially.
* Task 3: The sub tasks like breaking down of tasks, has to be done prior to implementation, hence this task cannot be done in parallel. Also, the burndown chart creation, will have to be done once the tasks are completed.

**BurnDown Chart : Sprint Duration 2 Days**

**Explanation / Analysis of the burndown chart** :

The ideal time line is explained in blue , and the actual timeline is in orange.

The initial time estimates of the tasks and the actual time they took are shown in the table below .

A screenshot of a computer screen

Description automatically generated

The actual hours for each of the tasks were different from the original estimates.

In the burndown chart, It is not possible to show the amount of effort put in , if it is different from the original estimate .

However, In the order to burn 70 minutes, it took more time . Hence we see some deviation of the actual timeline from the original one . Original timeline is a steep curve, which if I was accurate with estimation, the actual timeline would overlap with the ideal timeline .

There is one information that is missing however in the burndown chart.

If the total estimated hours is 240 hours, and if more or less hours were burnt, that information is not seen in the chart.

We notice that at hour 37, actual timeline and ideal timeline coincide, that is because perhaps some tasks took lesser time that estimated , so the time taken for tasks 1 & 2 which was higher that estimated, was compensated by task 3’s lesser effort required for completion.

It is difficult to analyse hour wise burndown chart . Since it was a small task with only 48 hours of sprint – it may have worked well. Otherwise, story point chart for a. longer sprint duration is much simpler.

**Swarming been helpful** ?

There was an impediment for the first task . This was due to different version of Java available on the system. And IDE not recognizing any one .

If there were multiple people on this project then, swarming in to find the root cause of this problem would have helped.

For other tasks, since they were fairly simple to achieve and did fall as planned, swarming was unnecessary for other tasks.

**Time Estimation Calculation based on the Feature 2 implementation for 1 Developer.**

A screenshot of a computer

Description automatically generated

**Functionality of the application**

**Feature 1** : Calculating the average based on the story points .

A screenshot of a computer

Description automatically generated

**Feature 2**:

The result for 2 developers for the sprint duration of 2 days.

A screenshot of a computer

Description automatically generated