RETROSPECTIVE

TEAM 8

## PROCESS MEASURES

### Macro statistics

Number of stories committed vs. done

**Committed stories:** 3

* Get ticket
* Next customer
* Call customer

**Done stories:** 2

* Get ticket
* Next customer

Total points committed vs. done

Committed: 13

* Get ticket: 5
* Next customer: 3
* Call customer: 5

Done: 8

* Get ticket: 5
* Next customer: 3

No. of hours planned vs. spent (as a team)

* Planned: 8 x 7 = 52
* Actual: 63,5 = 63.5

\*\*Remember\*\*a story is done ONLY if it fits the Definition of Done:

* Unit Tests passing
* Code review completed
* Code present on VCS
* End-to-End tests performed

> Please refine your DoD if required (you cannot remove items!)

### Detailed statistics

We considered 1 story point = 4 hours

| Story | Number of tasks | Points | Hours: estimation | Hours: actual |
| --- | --- | --- | --- | --- |
| #0 | 3 |  | 2.5 | 4.5 |
| Get Ticket | 7 | 5 | 20 | 40 |
| Next Customer | 4 | 3 | 12 | 11 |
| Call Customer | 4 | 5 | 20 | 12.5 |

* Hours per task average, standard deviation (estimate and actual):
  + Estimated:
    - Average: 17.33 ~ 17 hours
    - Standard Deviation: 3.77 ~ 3.7
  + Actual:
    - Average: 21.16 ~ 21 hours
    - Standard Deviation: 13.33 ~ 13.3
* Total task estimation error ratio: sum of total hours estimation / sum of total hours spent - 1:
  + - 0.18

## QUALITY MEASURES

* Unit Testing:
  + Total hours estimated: 5
  + Total hours spent: 8

No. of automated unit test cases: 5 (all tests run were automated)

Coverage (if available): 80%

* E2E testing:
  + Total hours estimated: 1
  + Total hours spent: 1
* Code review
  + Total hours estimated: 3
  + Total hours spent: 3

ASSESSMENT

* What caused your errors in estimation (if any)?
  + Initially we didn’t consider unit testing, code review, and learning new techstack.
  + We weren’t clear of the actual definition of done.
  + In terms of communication, we weren’t clear of the agenda of the meeting

(we didn’t explicit at the beginning of the meeting the goal of the meeting, the points to cover, the issues etc)

* What lessons did you learn (both positive and negative) in this sprint?
  + Positive:
    - Our stakeholders were happy with the work we delivered
    - Good team chemistry
  + Negative:
    - We could have completed more stories
    - Estimations could have been realistic
    - We need to work on dependencies of functional requirements and prioritize some tasks over others.
    - Initial planning, workload distribution
* Which improvement goals set in the previous retrospective were you able to achieve?
  + Not applicable
* Which ones were you not able to achieve? Why?
  + Not applicable
* Improvement goals for the next sprint and how to achieve them (technical tasks, team coordination, etc.)
  + Improve work definition, estimation, and self-assignment:
    - Defining and assigning all the tasks required in the definition of done
    - And self-assigning work accordingly
    - Be a little more realistic in terms of our estimations: we overestimated front end, we underestimated unit testing
  + Improving way of communication:
    - Keep the meetings short (have a meeting agenda) and strategize our discussion accordingly
    - Avoid talking about issues in the meeting
    - Avoid 1 to 1 communications in the meeting
    - Have small group work sessions so that people assigned to similar tasks can communicate better and real time
* One thing you are proud of as a Team!!
  + We have a really good tea and we have a really good chemistry. All of the people in this team are very nice. We are all very skilled and we share our knowledge.