Sprint Retrospective

Office Queue - Team #3

Demo overview

Infrastructure

Client

ReactJS

Server

Node.js
SQLite3
Express.js

Process

Stories, 1st sprint

1. As an **employee** I want to be able to notify the system that I can serve another request
2. As a **customer** I want to receive a ticket for my request so that I can perform my operation
3. As a customer I want to know the estimated waiting time so that I can eventually come back another time
4. As an **employee** I want to receive only customers performing specific operations so that I know how to help them
5. As an **office** manager I want to set up, start the system and configuring its parameters
6. As an office manager I want to visualize statistics so that I can eventually improve the office organization (counters, types of requests, etc…)
7. As a **customer** I want to be notified when it's my turn and to which counter I should go

Macro statistics

- 5/5 stories done
- 17/17 points done
- Total hours spent: 44
- Total number of tasks: 22
- Min number of tasks per story: 1
- Average number of tasks per story: 2.7
- Max number of tasks per story: 4
- Number of horizontal tasks: 10

Macro statistics

1) As an **employee** I want to be able to notify the system that I can serve another request **3 s.p.**

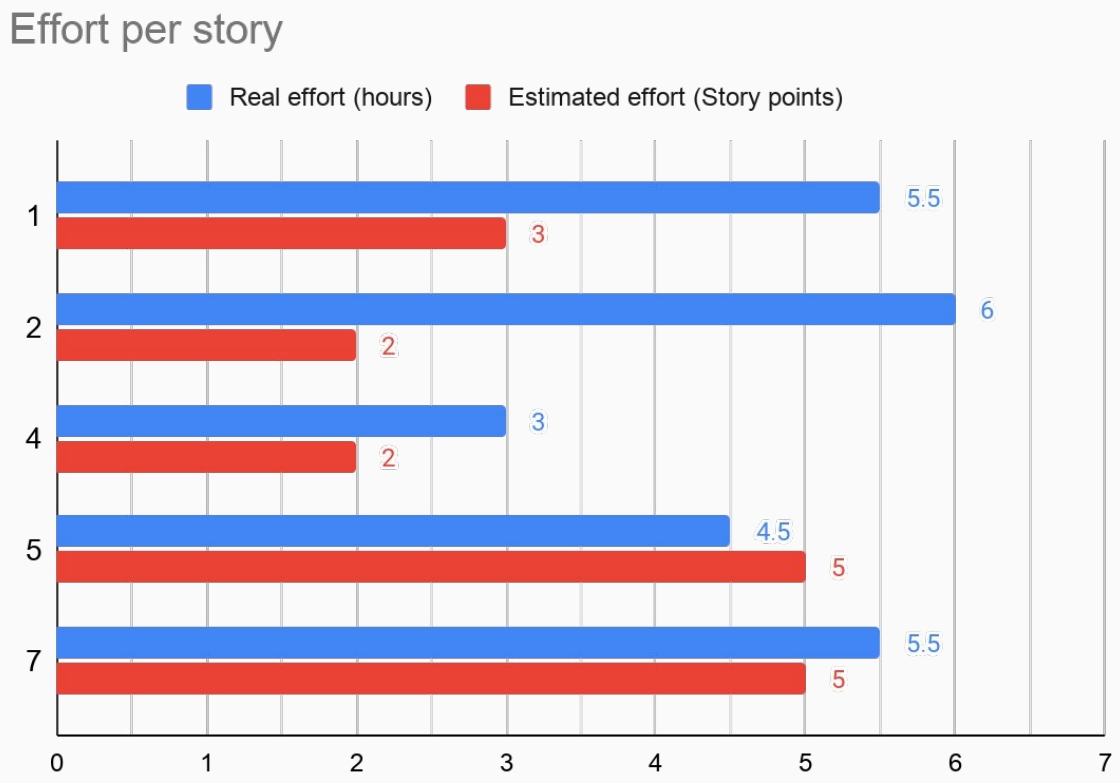
2) As a **customer** I want to receive a ticket for my request so that I can perform my operation **2 s.p.**

4) As an **employee** I want to receive only customers performing specific operations so that I know how to help them **2 s.p.**

5) As an **office** manager I want to set up, start the system and configure its parameters **5 s.p.**

7) As a **customer** I want to be notified when it's my turn and to which counter I should go **5 s.p.**

Longest story: 2
Shortest story: 4



Macro statistics

1) As an **employee** I want to be able to notify the system that I can serve another request **3 s.p.**

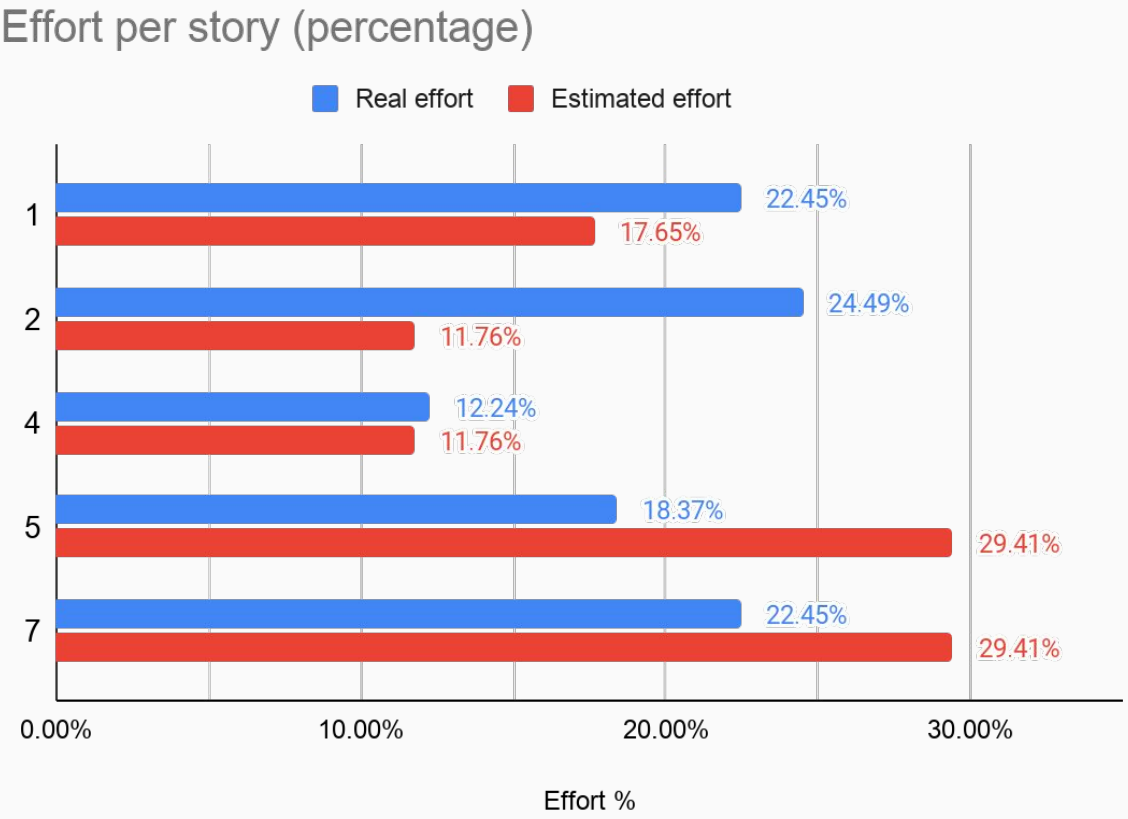
2) As a **customer** I want to receive a ticket for my request so that I can perform my operation **2 s.p.**

4) As an **employee** I want to receive only customers performing specific operations so that I know how to help them **2 s.p.**

5) As an **office** manager I want to set up, start the system and configure its parameters **5 s.p.**

7) As a **customer** I want to be notified when it's my turn and to which counter I should go **5 s.p.**

Longest story: 2
Shortest story: 4

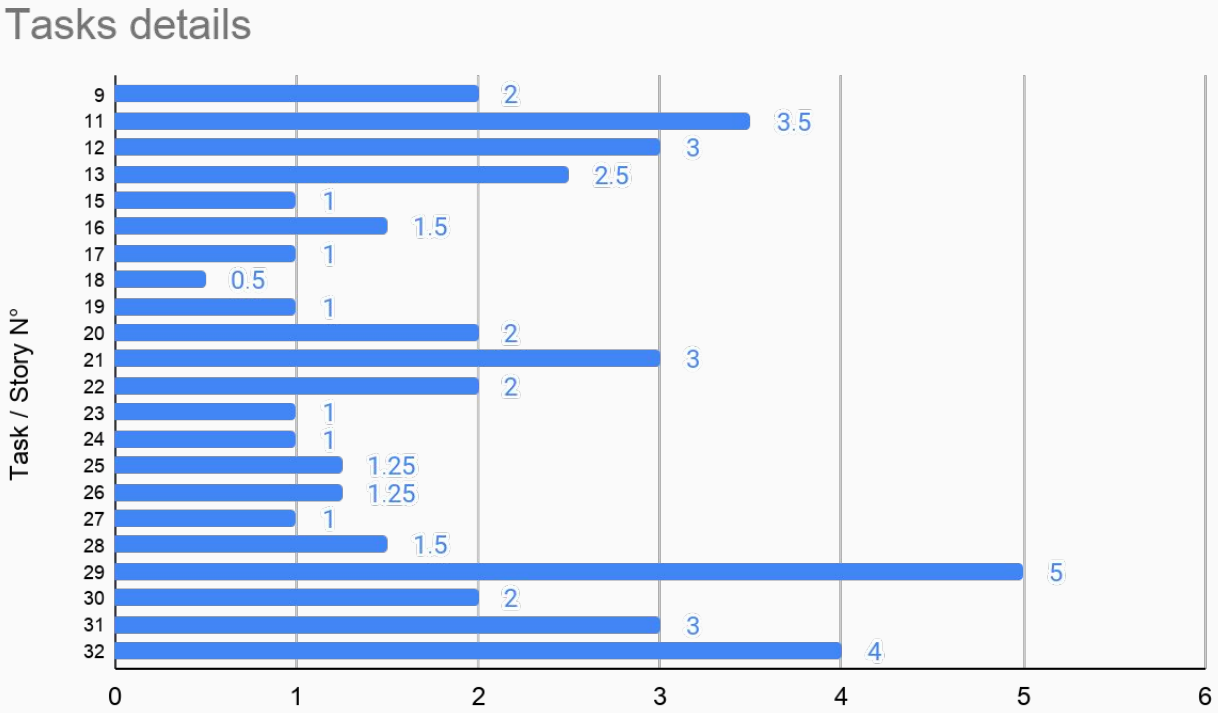


Detailed statistics

Average 2 hrs
Standard deviation: 1.16

Longest task: 29 “Study documentation”

Shortest task: 18 “[server] endpoint to insert list of possible operations”



Stories detailed statistics

ID	# Tasks	Points	Total hours estimation	Total hours spent
1	3	3	8.47	5.5
2	2	2	5.65	6
3		5		
4	1	2	5.65	3
5	4	5	14.12	4.5
6		3		
7	2	5	14.12	5.5

Total task estimation error ratio: Hours estimated / Hours spent = 48 / 44 = 1.1

Quality

Quality

- **We did not set up automated testing…**
  - We ended up spending some time in manual testing
  - We underestimated the amount of time required to do correctly automated unit testing and integration testing
  - Time spent: 3 hr (Unit + System testing)
- We did code review for each task
  - And some pair programming on the more difficult tasks
  - We did not estimate the hours needed for code review
  - Time spent: 2 hrs

Assessment

What caused your errors in estimation?

- We should have divided stories into tasks in a more precise way, investing some additional time in planning
- The low number of tasks made it harder to clearly separate the job among the team members, and since they were not so specific, it often happened that a task had “hidden” but complex sub tasks
- Some metrics have not been collected correctly or estimated before

What lessons did you learn (both positive and negative) in this sprint?

- It’s better to spend some additional time in sprint planning, because not defining stories/tasks in an exhaustive way can lead to confusion in the development phase
- We should have defined some kind of standard/convention at the beginning, because we realized afterwards that, since each member has different experience, it can be a problem for others (e.g. API naming convention, DB tables’ names, etc…)
- It’s also worth it spending some time in setting up (and documenting) the project structure in a clear way at the beginning, and constantly updating documentation, so that each team member knows what to do (and not just the one working on it)
- We need to plan much more time to write proper automated unit and integration testing

Propose 1 or 2 improvement goals for the next sprint and specify how to achieve them

1. Standardized code conventions
    a. Adopting a standard can allow us to make assumptions on how functions behave without reading documentation or analyzing code
    b. We lose less time fixing bugs caused by unexpected behaviors
    c. We will choose style guides and conventions for each technology we are going to use
2. Spend more time in planning and designing before writing code
    a. We can add other two steps after the GUI sketches/design
        i. Analysis of the user interaction with the GUI
        ii. Analysis of the necessary API calls to allow the user interactions in GUI

One thing you are proud of as a team!!

We are proud of having finished all the tasks we planned for this sprint, developing fully usable and working prototype, respecting more or less the total number of planned hours and splitting correctly the work according to each member experience and skills.

We are also proud of having received positive feedbacks from the stakeholders :)

Thanks!

Team 3

Galib Alili
Luca Antoniotti
Federico Bitondo
Giuseppina Impagnatiello
Dario Piazza
Luca Rinelli