26/11/24

Agile Retrospective

Team 5

- Alain Michel NITUNGA
- Elie Kheirallah
- Farouk Ait Oujkal
- Jamal Assou
- Léo Calvo Castaño

Team Assessement

Communication

What is going well?

- Discord
- Active participation in the project
- All team members constantly communicate

What areas could use improvement?

- More calls to view progress

What should we be doing differently?

- Discuss encountered bugs
- Share our new acquired knowledge so that everyone is productive
- Team must immediately communicate when a task cannot be achieved on time

Team velocity and planning status

- Shared repositories between team members to effectively distribute tasks.
- Our team is constantly planning who accomplish which **User Story,** and when it will be done, to work effectively and avoid any conflicts in our Github repositories.
- Team members are often well delivering on their tasks on time.

Sprint Review

What is going well?

- Taking notes of customer feedback.
- Our progress seems to be on track with the customer's expectations.
- Getting customer feedback is helpful.

What areas could use improvement?

- Prioritizing more important tasks over others

What should we be doing differently?

- Continuous Training: Invest in continuous training for team members to improve technical and project management skills.

Planning and distribution of tasks

What is going well?

- We have consistently been able to complete tasks within the deadlines, even though sometimes they were not delivered to the correct place or in the expected format.

What areas could use improvement?

- Some tasks or ideas are anticipated by certain team members but not explicitly shared with the group in advance. This can lead to implicit expectations or imagined deadlines that are not communicated.
- As a result, other team members may feel caught off guard when they are assigned tasks related to topics they hadn't previously considered.
- There is also a tendency to focus on the more appealing or rewarding tasks while delegating more challenging or less desirable work to others.

What should we be doing differently?

- After each meeting with the client, we should immediately assign tasks for the upcoming week. This will ensure:
  - Everyone is present for task distribution.
  - Decisions are made while project updates and client feedback are still fresh in our minds.
  - Workload distribution is **fair** and agreed upon **collaboratively**.

helb Ilya Prigogine

Understanding Business and **Technical** Requirements

At this point, we've clearly understood what to do and where to it especially regarding the need for real-time updates (**SignalR**), the Azure Function.

What Areas Could Use **Improvement**?

Based on the last **feedback** of the client on the JSON structure of the app, he said that he wanted that we restrict the users from seeing other users' data and improve the user experience on the app.

We should have Drawn a mockup of the front-end and present that to the client during the sprint to validate it based on his expectations before its implementation.

Technical Knowledge (Code, ...)

Despite initial unfamiliarity with the technologies and some sections that were missing on Azure Portal, the team implemented key features such as Azure functions, SignalR, and developed in the Angular framework.

What Areas Could Use **Improvement**?

The team faced challenges with **Azure Functions** , especially Event grid, and **SignalR** as we had limited prior experience with these technologies.

What Should We Be Doing **Differently**?

Refactor some code to use the technologies more efficiently

The use of **JIRA, Teams and Structurizr**

On a positive note, we're documenting everything on JIRA, EPICs, user stories and for each sprint, we're clearly mentioning what must be done, the priority of the task, and who is in charge of it.

Teams is being used mainly for depositing tasks and communicating with the client and Structurizr has been used for the Container structure

What areas Could Use **Improvement**?

We should have done differently the Jira's Tasks management to avoid tasks that are remaining **in progress from last sprint 'till now. (**Limit the WIP**)**

Product Assessement

Are we meeting the customer's expectations and needs?

- Customer's needs are being mostly met on time.
- Functionalities delivered so far:
  - Web App in Angular TS (frontend) / C# (backend)
    - Login, register and profile with MongoDB
    - Clients can add their homes, rooms and devices
    - Clients can toggle their devices (ON/OFF)
    - Real-time updates in the Web App from Server -> Client
    - Development in progress in the Azure Function

- 3 tasks are slightly late due to multiple technical challenges:
  - JWT Tokens
  - Bi-directional Real-time connection with the Azure function and the Web App is being challenged by CORS policies
  - Turning HTTP Trigger to an Event Grid

Reliability

Is our app reliable?

Pros:

- Real-time connection between the Server and the Client within the web app itself appears to be reliable and working smoothly.
- The web app is quick and easy to use, no errors are left in the console.

Cons:

- JWT tokens are not yet implemented, anyone can request anything to the server.
- Some security concerns, such as directly inserting secret keys in the code source could be prevented using additional technologies.

Quality and Efficiency

What is going well?

- The project is on schedule, with milestones being met on time.
- Every Sprint and meeting with the client is aligning us with his demands.

What areas could use improvement?

- Security: Strengthen the system's security by implementing robust features like user authentication and data encryption (e.g., JWT tokens) to build trust.
- Code Quality: The source code of the web app needs refactoring to improve maintainability. This includes:
  - ➢ Removing redundant code to reduce duplication.
  - ➢ Reorganizing folder structures for better clarity.
  - ➢ Standardizing naming conventions to enhance consistency.

What should we be doing differently?

- Prioritize technical debt: Dedicate specific time to refactoring and improving code quality during each sprint to ensure long-term maintainability.

helb
Ilya Prigogine

User Experience

What is going well?

- The core functionality of turning lights on and off remotely is intuitive and responsive.
- The interface has consistent navigation, making it easy for users to find features. (In Progress)

What areas could use improvement?

- The design is functional but lacks modern aesthetics that could enhance user engagement.
- The frontend navigation requires additional work to make transitions between multiple pages more intuitive.

What should we be doing differently?

- Collect more customer feedback and involve a diverse group of users to improve the overall user experience.

helb
Ilya Prigogine

HOME
WORK
classroomclipart.com
http://classroomclipart.com

🚶 …Our **Experience** With This Project

- From the beginning, none of us were familiar in Azure Functions or SignalR which presented initial challenges.

- Each of us had a **role** to play, either focusing on the backend or the frontend, and we documented our progress on JIRA to keep track of what we were doing.

- While we encountered some obstacles, such as unexpected technical issues and tasks taking longer than anticipated, we worked collaboratively to overcome them. Whenever one **team** member faced a challenge, other **stepped in** to offer assistance, ensuring we stayed on track. breaking tasks into smaller parts before starting the implementation would have helped us manage the complexities more effectively.

- We planned our tasks early, continuously adapted to new challenges and prioritized delivering as much value as possible in each Sprint.

The bad:

- Frequent presentations proved to be more time-consuming than we initially expected, but they also allowed for continuous feedback, which is one of Agile's key strengths, **guaranteeing** that we stayed **in sync** with the client's evolving needs.
- We learned that working in a team involves different people with different values, different levels of motivation and ways of working, which can lead to multiple conflicts throughout the project.

Lesson learned:

- Collaborating in a team, with diverse skills and perspectives, can be challenging. But through our **differences**, we learned how to find solutions to grow stronger together as a team, and we pushed each others to become more pro-active and offer more initiatives, start our tasks early to have the opportunity to ask a lot of technical questions on time which will guarantee our success.

Lessons learnt

Transparent Communication

- Always **communicate** what has been completed, what remains pending, and any challenges faced. This ensures team members are aligned and can **support** each other effectively.

Customer Feedback Drives Progress

- Regularly present functional **deliverables**, even if incomplete, to customers. This helps evaluate **alignment** with their expectations and facilitates clear, actionable feedback.

Plan and Assign Tasks Promptly

- Assign tasks immediately after meetings to ensure clarity, fairness, and alignment while feedback and updates are fresh.

Focus on Code Quality

- Regular code refactoring is a must to enhance **maintainability** and **reliability**

Team Learning and Adaptation

- Encourage shared learning to adapt to unfamiliar technologies and foster a collaborative environment that builds **resilience** and **technical capability**.