Condospective

Introduction

Firstly, coming into this sprint, our team's focus was to focus on testing to achieve at least 80% of code coverage. To do so, we collectively agreed to put aside the implementation of new features and dedicate more time and effort on back-end and front-end unit testing the increase. Furthermore, based on the sprint #2 feedback, a deployment diagram was added to our system architecture document.

What went wrong

1- Transition of Front-End Tools

One of the more challenging parts that was found again in this sprint was the change of tools done in sprint#2 for our front-end part. Indeed, our team encountered multiple issues and limitations with the Uno Platform, initially chosen, such as compatibility issues, performance concerns, or lack of community support. Therefore, our team decided to choose a different route and utilize React instead for the continuation of the project.

However, the shift in tools necessitated the rewriting of code, and the adjustment of development processes, resulting in a temporary slowdown of project progress. While many of these concerns were resolved during sprint#2, many members of the team are still accommodating to this transition. To address this challenge, our team recalibrated project milestones, revised timelines and allocated additional resources to facilitate the transition to React.

This transition could have been avoided by choosing the right tool for all team members at the beginning of the project. To do so, our team could have conducted a more thorough evaluation of the potential tools' considering factors such as platform compatibility, learning curves, long-term maintenance requirements, integration capabilities, etc.

2- React

An aspect that went wrong once again for some, but not all team members, was the utilization of React for our front-end. Since React introduces new concepts such as JSX syntax, virtual DOM, and component-based architecture, it was challenging for some team members that haven't had hands-on experience with it.

The complexity of React concepts led to confusion, frustration, and slower adoption among team members, affecting productivity and overall motivation. To address the issue, the team provided comprehensive documentation and tutorials to familiarize team members with React concepts gradually.

Implementing a structured pair programming where experienced React team members are paired with those new to React could have facilitated this issue for the whole team.

3- C# experience

A challenge that the team continued to deal with during this sprint is the lack of experience of some team members with C#, used for this project. This occurred due to factors such as the limited exposure to C# in previous projects and the limited time assigned to achieving the deliverables.

This aspect has led to slower adoption and learning curves throughout sprint #3, especially with addition of test cases. To fix this issue, the team continues to leverage online resources and documentation to provide additional support. Furthermore, the team has assigned tasks based on team members' strengths and areas for growth, allowing for a gradual hands-on experience.

Improvements could have been made by conducting a skills assessment at the outset of the project to identify the gaps within the team. Based on that assessment, the choice of language used for this project could have differed.

4- Documentation

An aspect that our team struggled with during this third print, but not the first two, was not completing the documentation at the last minute and before the deadline imposed.

This was caused by the high-pressure and extreme focus of successfully achieving 80% code coverage on time.

By focusing on testing, team members neglected the importance of the documentation aspect of our sprint and felt compelled to leave it to the last day. To address this issue, the team agreed to find a balance between achieving the testing or the implementation of new features and the writing of documents. To do so, we've decided to include specific days where every member of the team would work on the documentation rather than implementation/testing.

Finally, setting realistic and achievable deadlines based on thorough project planning and estimation could have prevented excessive pressure on team members and allowed for a more balanced between implementation/testing and completion of documentation.

5- Jira

The use of the Jira Platform was not as efficient during this sprint as it was during sprint#2. Despite its capabilities, we fell short in maximizing its benefits for organizing tasks, assigning responsibilities, and monitoring progress effectively throughout the development cycle.

Acknowledging Jira's value to our team, we understand that our usage during this sprint didn't reflect its full potential. To address this issue, we've agreed to provide more detailed task descriptions and acceptance criteria in future sprints to improve clarity and streamline our processes effectively.

This issue could have been avoided by setting frequent reminders for every member of the team to participate in Jira and to hold them accountable if they haven't shown consistent participation.

What went right

1- 80% Code Coverage

In this new sprint, our team had effortlessly achieved an impressive 80% code coverage, marking a significant milestone for our team. The implementation of necessary tests that were not achieved in prior sprints was seamless, reflecting our team's proactive approach to quality assurance.

By ensuring 80% code coverage, we can now state that are our project is well-protected against undetected bugs and regressions. This ensures that our product maintains its high quality and reduces the need for extensive maintenance efforts in the future. To reach this level of code coverage, our team focused on prioritizing testing and allocating dedicated time and resources effectively to address any potential gaps in our testing strategy.

However, this achievement could have been reached in earlier sprints if our team hadn't underestimated the importance of testing in our project. Testing should have been a priority over the implementation of new features from the start of the project. Moving forward, we will continue to uphold this standard of excellence in testing, leveraging our newfound success as motivation to maintain and even surpass our current levels of code coverage.

2- Continuous Monitoring

An aspect that continued to work well during the development is the continuous monitoring and adjustment. Regular meetings and updates on task progress helped keep everyone aligned and focused on their respective responsibilities once again. This included reallocating resources, re-prioritizing tasks, or providing additional support to team members as needed.

By doing so, our team was able to adapt quickly to evolving requirements, priorities, and challenges. It enhanced our ability to stay aligned with project goals and respond to the bigger changes that were made in the project environment. To address this aspect, daily stand-up meetings were conducted. These meetings have provided an opportunity for team members to share updates, discuss progress, and raise any issues or concerns.

However, the integration of monitoring tools with automated incident response systems could have helped us more. By automating the response to common alerts and predefined scenarios, our team could have reduced manual intervention, minimized downtime, and improved overall system reliability.

3- Agile Methodology Implementation

The implementation of Agile methodologies remained something that worked well during the development. This aspect occurred due to clear understanding and commitment from the team. Agile principles such as iterative development, frequent feedback, and adaptive planning were embraced and integrated into the team's workflow.

The impact of effective Agile methodology implementation included increased productivity within the team, as well as improved collaboration among all team members. To ensure this implementation, the team established daily stand-ups and sprint planning meetings. Furthermore, regular retrospectives were held to keep up with the team's processes and identify areas that needed more work.

However, this implementation could have been better by putting more effort in strengthening communication channels between the different parts of the sprints. For example, if some members finished earlier than anticipated, and other members needed help with their parts, teammates should be able to move from part to part to help each other. This aspect is something our team still must work on.

4- Good Pace

One aspect that continued to work well during the development was the pace at which our team worked during the sprint. This good pace was reached by the effective sprint planning and task breakdown our team maintained from the first print. This time around, the prioritizing and the assignment of tasks was done more quickly.

The good pace in executing tasks resulted in increased productivity, allowing our team to deliver more value within the sprint timeframe. This positively impacts overall project progress and contributes to meeting project milestones and deadlines. To address this aspect, our team prioritized tasks based on their importance and urgency, focusing on high-priority items first to ensure maximum value delivery within the sprint timeframe.

While the good pace in executing tasks is commendable, there is always room for improvement. Indeed, when it comes to the implementation of code, identifying and mitigating risks concerning the implementation could've avoided the large amount of time that was spent on resolving coding issues on the back-end part.

5- Collaborative Problem Solving

An aspect that did work well and was maintained during the development of sprint #2 is our team's collaborative problem solving. This was reached, because each member of the team brought unique perspectives and problem-solving approaches to the table.

This aspect increased creativity, faster problem resolution, and a stronger sense of ownership and accountability among team members. To address collaborative problem solving, the team organized brainstorming sessions and workshops to generate ideas collaboratively.

However, for future sprints, our team could organize more opportunities to share knowledge and expertise such as in-person meetings to further enhance this aspect.

Conclusion

In conclusion, the team has encountered both achievements and challenges in equal measure during this third sprint. One of the main takeaways from this sprint is the importance of testing and consistently reaching at least 80% code coverage. Despite facing obstacles such as learning curves of front-end tools and lack of use of the Jira platform, the team's ability to work together, share knowledge, and support one another has been instrumental in overcoming obstacles. Additionally, the team has continued to gain a deeper understanding of the technologies that were retained from sprint #1. However, when it comes to the new technologies that were installed in sprint#2, there is always room for improvement and growth for some team members as this project continues. Finally, this sprint has highlighted the significance of prioritizing testing and not underestimating it when having to implement new features. As the team navigated through unfamiliar territories implemented in this sprint, all team members demonstrated their ability to adapt quickly and remain focused on the project's requirements and deadlines to respect.