OffbeatEscape

Analysis of Management

University of Windsor

Objective

This Analysis of Management Document shall explain how the global management plan was applied to the OffbeatEscape project. The evolutions made, overall performance and impact of the plan on the project.

Author –Team 4
Recipient - Dr. Aznam Yacoub
Updated On - 13-04-2021

Index

SDLC Analysis

- Lifecycle was correctly followed at all points and the rules were kept in check. Any backlog for a sprint was moved to next with highest priority.
- The final project adhered to initial proposal and all the major client's requirements were fulfilled within the time frame of 4 sprints of length 2 weeks each.
- Despite a major change/ update of client's requirement in 3rd Sprint for the most significant feature of the project, team was able to stretch through it and stick to prior committed timeline for product delivery.
- The only backlog for the project were additional functionalities of Chatbot and

MFA. Following steps were performed for every sprint:

Requirement Gathering

At the beginning of each sprint, a client call was arranged to discuss last sprint's analysis and
confirm requirements for next sprint features/tasks.

Requirement Analysis

Client's feedback from requirement gathering phase was analysed and in case of any discrepancies, another meeting was scheduled to close the open points.

Specifications

Specifications for all the features was shard with other team members for their inputs, suggestions, and improvements.

Design

Design was created once the specifications were approved by the team. Design and research for task to be implemented went hand in hand given knowledge constraints and inexperience with tech being employed.

Development

Development was done based on specifications and design. Code Reviews and walkthroughs were done to ensure quality adherence. Documentation was added parallelly to help a new member or a co-member to get the overview and pick up with development quickly when needed.

Testing

With requirement analysis, testing started in parallel to analyse the requirement, plan the test suite, cases, and design.
All the features developed in one sprint were put to testing in next and the bugs were reported over Jira with added details for steps and screenshots for clear understanding of dev team.

Deployment

Starting from Sprint-2, deployment was researched, and setup was being progressively performed and achieved by the end of project.

Management & Risk Plan Analysis

Sprint Planning

Features for a sprint were decided based on meeting with the team. Task priority & client's feedback were considered while deciding goals for a sprint. Team's availability, knowledge constraints & previous sprint's analysis were considered while dividing tasks among the members.

Conflict Resolution

In case of an internal conflict, the involved members were advised to talk it out and explain their point of views for a clear picture of the problem. In case of indecision, either voting was done to conclude, or the problem was phrased as a question or suggestion to Product Director and his response aided the final decision.

Resource Unavailability

In case a resource was unavailable, the task was picked by someone else with buffer on team. Tasks were planned during sprint plan ensuring the knowledge of unavailability of a team member during the sprint.
In case of unplanned unavailability, backup development resources would help in with development.

Code Reviews and Walkthroughs

Code was pushed to GitHub and pull requests were raised to ensure peer reviews are performed before moving any code to main branch.
Code walkthroughs were done weekly for the first 2 sprints to ensure the flow of application and code is clear to all the team members and to ensure the same can be used as learning point for members not used to the technologies in action.

Trainings

Self-learning through LinkedIn Learning and other free documentations was encouraged. Code walkthroughs conducted for first 2 sprints also aided the process.

Quality

The inputs from Daily Scrum meetings , milestone meetings and client feedbacks were incorporated to make sure we are developing a product with good quality .
Automation was added to avoid retesting of basic features that were put to retest after every feature was delivered to QA Engineer.
All the bugs reported were priorities by QA Engineer and Development team to ensure a quality build is delivered to client.

Evolution

SDLC

1. For Requirement Gathering & Analysis, the rule was defined to get the initial requirement meeting with closed in first 3 days of sprint itself to avoid delays in development due to dependencies on pending requirements.
2. For Deployment, research and process should work parallelly with the sprint development plan was incorporated going forward from Sprint-2 to keep that in track as well.

Management

1. Resource unavailability whether informed or uninformed was not defined in the SQAP initially but plays an important role when planning a project.
2. Sprint Planning was limited to a meeting with team and the other factors affecting the decisions were not clearly defined which were later put in place from Sprint-2 onwards.