**LESSON LEARNED REPORT**

**Project: AI Customer Analyzer**

**Group: 19**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Release: MARCH 2023 | |  | | |
| Date: 15 MARCH 2023 |  | | | |
|  | | | |  |
| **PRINCE2** | | | |  |
|  | | | |  |
| Author: Gunarakulan Gunaretnam (Project Manager) | | |  | |
| Owner: Dr. Yasas Jayaweera (Project Executive) | | | |  |
| Client: DreamSpace (Private) Limited  Project Manager: Gunarakulan Gunaretnam (2208408)  Startup Manager: Sangeetha Thangavadivel (2135801)  Risk Manager: Haritha Thavarajah (2211320)  Schedule Manager: Mathumitha Arasakulasoorian (2211336)  Quality Manager: Delaxsan Raj Sathiyanesan (2211294) | | | |  |
| Document Ref: Lessons Learned Report Brief Group 19 | | | |  |
| Version No: 1.0.0 | | | |  |

# 1 Lessons Learned Report History

## 1.1 Document Location

This document is only valid on the day it was printed.

The source of the document will be found on the project's PC in location

## 1.2 Revision History

**Date of this revision:**

**Date of Next revision:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision date** | **Previous revision date** | **Summary of Changes** | **Changes marked** |
|  |  | First issue |  |

## 1.3 Approvals

This document requires the following approvals.

Signed approval forms are filed in the Management section of the project files.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Signature** | **Title** | **Date of Issue** | **Version** |
| Dr.Yasas Jayaweera |  | Project Executive |  |  |
| Gunarakulan Gunaretnam | A picture containing text, hydrozoan, night sky  Description automatically generated | Project Manager | 15-03-2023 |  |
| Kishoth Navaretnarajah | Shape  Description automatically generated | Client | 15-03-2023 |  |

## 1.4 Distribution

This document has been distributed to:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Date of Issue** | **Version** |
| Gunarakulan Gunaretnam (2208408) | Project Manager | 15-03-2023 |  |
| Sangeetha Thangavadivel (2135801) | Startup Manager | 15-03-2023 |  |
| Haritha Thavarajah (2211320) | Risk Manager | 15-03-2023 |  |
| Mathumitha Arasakulasoorian (2211336) | Schedule Manager | 15-03-2023 |  |
| Delaxsan Raj Sathiyanesan (2211294) | Quality Manager | 15-03-2023 |  |

# 2 Table of Contents

[1 Lessons Learned Report History 2](#_Toc131015402)

[1.1 Document Location 2](#_Toc131015403)

[1.2 Revision History 2](#_Toc131015404)

[1.3 Approvals 2](#_Toc131015405)

[1.4 Distribution 2](#_Toc131015406)

[2 Table of Contents 3](#_Toc131015407)

[Lessons Learned Log 4](#_Toc131015408)

[3 Purpose 4](#_Toc131015409)

[4 Management/Quality Process Assessment 4](#_Toc131015410)

[4.1 Went well: 4](#_Toc131015411)

[4.2 Went badly: 5](#_Toc131015412)

[Lacking: 6](#_Toc131015413)

[5 Deviations 6](#_Toc131015414)

[6 Method/Tool Performance 6](#_Toc131015415)

[6.1 Methods: 6](#_Toc131015416)

[6.2 Tools: 7](#_Toc131015417)

[7 Project Issues 7](#_Toc131015418)

[8 Recommendations 8](#_Toc131015419)

[9 Measurements of Effort 8](#_Toc131015420)

[10 Statistics 9](#_Toc131015421)

# Lessons Learned Log

## 3 Purpose

This document implies the lessons the project team learned throughout the project process. This includes the information regarding the project process management to give the outcome. The tools and methodologies used, issues faced by the project team, and how each member rectified the problems to improve the development standards of the project.

## 4 Management/Quality Process Assessment

The management and quality processes for the project were moderately successful; some of the tasks went well, some went severely, and some were lacking. The following is an assessment of the critical management and quality processes:

### 4.1 Went well:

* **Project initiation:** The process went well because it was well-planned and implemented in a price manner with a clear understanding of the project objectives, scope, and deliverables.
* **Stakeholder management:** During the development and implementation process, our team effectively engaged and managed stakeholders, helped maintain their support, and met their needs and expectations.
* **Artifact Design:** One of the most significant pillars of every project is the architecture of artifacts; during this stage design of artifacts designing went well, our team created well-designed artifacts, including the project charter, requirements document, design specifications, and test plan, and the artifact was developed according to the defined standards, it was revived and approved by the project's stakeholders. Our team used various tools and techniques to build the artifacts, such as templates, document management systems, and version control software. These tools ensured the artifacts were consistent, well-structured, and easy to maintain. Our team also followed a collaborative approach, where multiple team members contributed to the development of the artifacts.
* **Development (Coding):** The coding development process appeared well. The development team completed the coding tasks within the designated time frame and budget. The team utilized appropriate coding methodologies and standards, resulting in a high-quality codebase. Our team also worked collaboratively and effectively, ensuring that any issues or bugs were promptly addressed and resolved. Regular code reviews were conducted to ensure the code was maintainable and scalable and met the project requirements. The development team also utilized appropriate version control systems, allowing for easy collaboration and management of code changes.
* **Startup Aspects:** To provide better work, the startup team efficiently organized the necessary resources, including personnel and equipment. We established a clear roadmap and worked collaboratively to ensure the project stayed on track and met its objectives. Moreover, the startup team demonstrated high adaptability and flexibility, which was crucial in handling unexpected challenges during the development process.

### 4.2 Went badly:

* **Communication management:** Communication is one of the hardest things in business management when it comes to clients; many possibilities are there to be misunderstood because of the lack of technical skills of the client; our team also faced some communication issues with clients, such as misunderstanding their requirements, this leads us to delay in development.
* **Change management:** Our project team struggled to manage changes to the project scope and requirements during the change management, which led to confusion and rework. A more robust change management process could have helped to minimize these issues.
* **Quality assurance:** The quality assurance process was not fully implemented, which resulted in some deliverables that did not meet the expected standards. More rigorous quality assurance processes could have helped to ensure that all deliverables met the required quality standards.

### 4.3 Lacking:

**Resource management:** Our project team could have more effectively managed resources, which resulted in some team members being overloaded while others needed to be more utilized; we also wasted so much time preparing unwanted documents that were optional.

## 5 Deviations

* Unexpected bugs: Many unexpected bugs discovered during the testing phase delay the project and require additional resources.
* Scope creep: Clients or stakeholders requested additional features and functionality not initially specified in the project; this increased our workload and affected the project timeline.
* Delays in third-party dependencies: The project relied on third-party software. There were some delays or unexpected issues with these dependencies.
* Personnel changes: Some health issues happened to our team members that caused a delay in development, and more attention was needed from other developers.
* Data loss or corruption: Data loss or corruption can cause significant delays and may require the development team to re-create lost data or revert to an earlier software version.

## 6 Method/Tool Performance

### 6.1 Methods:

* Agile development methodology was used, which enabled the team to adapt to changing requirements and prioritize work items effectively.
* Continuous integration and continuous deployment (CI/CD) pipeline were set up, allowing for automated building, testing, and software deployment, resulting in faster release cycles and fewer errors.
* Object-oriented programming (OOP) was used, which facilitated code reusability and maintainability.

### 6.2 Tools:

* Version control systems (VCS) such as Git was used, allowing for collaborative development, tracking of changes, and easy codebase management.
* Issue tracking system such as Azure Board was used, which enabled the team to track bugs, tasks, and feature requests, ensuring that work items were addressed promptly.
* Automated testing tools such as Selenium was used, which helped in testing web applications, detecting defects, and reducing the time and cost of testing.

## 7 Project Issues

* Communication Gaps: There were communication gaps between the development team and stakeholders. The stakeholders did not provide timely feedback, which delayed the project timeline. Additionally, there were instances where the development team needed to communicate more e effectively, resulting in misunderstandings and errors.
* Integration Challenges: The project involved integrating multiple systems, leading to challenges. The integration process was complex, and the issues were ensuring the different systems worked seamlessly together; there are three parts of this system as vision monitoring part, the audio monitoring part, and the management panel; these all connected to work as a single system, this could be the biggest challenge for us.
* Resource Constraints: The project faced resource constraints, which impacted the quality and timeline of deliverables. The development team faced challenges meeting project deadlines due to lacking skilled resources.

## 8 Recommendations

* Clearly define project goals and objectives: To avoid scope creep and ensure that the project stays on track, it is essential to describe it. This will help ensure that everyone on the team understands what is expected of them and what they need to deliver.
* Use agile project management methodologies: Agile methodologies are well-suited for software development projects. They allow for greater flexibility and adaptability and are particularly effective in managing complex projects with changing requirements.
* Improve communication among team members: Communication breakdowns can lead to delays, misunderstandings, and other project issues. Therefore, it is essential to establish clear lines of communication among team members and encourage open dialogue throughout the project.
* Invest in training and development: To ensure that team members have the necessary skills and knowledge to manage and execute the project effectively, it is essential to invest in training and development programs. This will help to build team members' capacity and improve overall project performance.
* Use project management software: Project management software can help streamline project processes, improve communication, and enhance collaboration. Consider using software tools such as Jira, Trello, or Asana to improve project management efficiency and effectiveness; we use Azure Board as a project management tool.

## 9 Measurements of Effort

* Lines of code (LOC): This metric measures the number of lines of code written for the software. It is a good indication of the amount of effort required to create the software. However, it is essential to note that LOC is not always an accurate measure of productivity.
* Function Points (FP): This metric measures the software’s functionality based on the user's point of view. It is a better measure of productivity compared to LOC.
* Person-hours: This metric measures the total hours the team works to complete the project. It considers the number of team members, roles, and project duration; our team used Clockify, an online time-tracking tool.
* Defect density: This metric measures the number of defects or errors found in the software. It is a good indicator of the software’s quality and the effort required to fix the flaws.
* Schedule variance: This metric measures the difference between planned and actual schedules. It is a good measure of the project plan’s accuracy and the effort required to complete it on time.

## Statistics

During the testing phase, 20 errors were identified and corrected. This indicates that the testing process also effectively identified and fixed issues before the product was released, building UI/UX in Python PyQt caused a lot of errors.