Software Development Life Cycle for Leckie’s Choice

Summary

This report evaluates the most suitable software development life cycle for Leckie’s Choice (LC) based on its requirements, uncertainties, and the capabilities of various life cycles. The Agile Development life cycle is recommended as the most appropriate due to its flexibility and iterative nature. The Iterative life cycle is considered the second most suitable, while the Waterfall model is deemed least appropriate due to its rigidity and lack of adaptability.

Introduction

Leckie’s Choice (LC) have grown from a small business into a significant distributor, wholesaler, and retailer of electrical equipment and services. With the business environment becoming increasingly competitive, and the current operational model showing signs of strain, LC needs to embrace modern technology to stay viable.

Problem Statement

LC is experiencing a decline in sales, increased customer queries, and competition from wholesalers offering better promotions and faster service. The company needs to modernize its technology to streamline operations, enhance customer experience, and regain market competitiveness.

Objectives

- Implement a system that supports online order submission.
- Develop capabilities for maintaining customer loyalty and promotions.
- Enable customer profiling and purchase history tracking.

Requirements of Leckie’s Choice (LC)

1. Online Order Submission: A system for customers to place orders online.
2. Customer Loyalty and Promotions: Features to manage promotions and track customer loyalty.
3. Customer Profiling and Purchase History: Tools for tracking and analysing customer behaviour and purchase history.

These requirements are high-level and may evolve based on customer feedback and market changes. Continuous validation with customers is essential to ensure the system meets their needs effectively

The least suitable life cycle: Waterfall model

The Waterfall model is the least suitable life cycle for LC's case study due to its rigid, linear structure that doesn't accommodate changing requirements or provide early customer feedback. This model poses high risks, as testing occurs late in the process, potentially leading to costly issues. Its long development cycles and inability to support incremental delivery make it a poor fit for LC, which needs to adapt quickly to evolving market demands and technological advancements.

The second most appropriate life cycle: iterative life cycle

In the Iterative life cycle, each incremental build results in a fully functional software system, enabling customers to observe progress as soon as the first iteration is completed. This approach is particularly effective for adding new features to an existing system. It supports parallel development, allowing multiple features to be developed simultaneously. Additionally, testing is streamlined because tests can be tailored specifically for each iteration.

The First Life Cycle: Agile Development Life Cycle

The Agile Development life cycle is characterized by its iterative process, flexibility, and focus on customer collaboration. For LC, these qualities are crucial in addressing the rapidly changing market demands and technological advancements. The followings are the advantages of the Agile Development.

- **Quickly access minimum viable product**: Agile Development will deliver MVP at the first stage, which can help develop team to get customer feedback as soon as quickly.
- **Flexibility**: Agile Development allows develop teams to adapt changes quickly. As new customer feedback is received, the development team can adjust the project scope, ensuring that the final solution meets evolving needs.
- **Iterative Progress**: By breaking down the project into smaller, manageable iterations, LC can see tangible improvements after each sprint. This incremental delivery helps in identifying and resolving issues early, minimizing risks.
- **Customer Collaboration**: Agile emphasizes close collaboration with stakeholders.
- **Focus on Quality**: Regular testing and reviews within each iteration ensure that the developed solutions are of high quality, reducing the likelihood of post-deployment issues.
- **Continuous Improvement**: Agile iterative nature encourages regular reflection and adjustment, fostering a culture of continuous improvement within LC.

Agile Development generally adapts to changes through iterative cycles, ensures quality through repeated testing, and enables the project team to maximize the system's value by obtaining continuous customer feedback.