

Approaches to Implementing the SDLC in Developing an E-Commerce Platform

Course Title: CS502052 – Enterprise Systems Development Concepts (ESDC)

Instructor: Phuc H. Duong

Background

An e-commerce company, **ShopSphere**, aimed to revamp its existing platform to handle growing traffic, improve customer experience, and integrate advanced features like AI-driven recommendations. The company explored two different approaches for developing its platform: the **Waterfall model** and **Agile development**. Both approaches followed the Software Development Life Cycle (SDLC) framework, but they significantly impacted the project's execution, flexibility, and outcome.

Objectives

This case study examines the implementation of both the Waterfall and Agile models in ShopSphere's e-commerce platform development project, highlighting the strengths and weaknesses of each model in relation to the project requirements, execution, and final outcome.

Approach 1: Waterfall Model

The Waterfall model follows a sequential design process where each phase must be completed before moving to the next. For ShopSphere, this approach was initially appealing due to its structured nature and clear timelines.

Phases in Waterfall for ShopSphere:

1. Requirements Analysis:

- ShopSphere's business team spent two months collecting requirements, including feature specifications, customer needs, and technical infrastructure upgrades. This phase was exhaustive, ensuring that all details were outlined before moving forward.

2. System Design:

- After requirements were defined, a team of architects designed the system. This phase lasted another three months. The design documents were comprehensive, covering the database structure, system architecture, UI/UX, and back-end integration.

3. Implementation:

- The development team began building the platform based on the detailed design documents. Coding took four months, as each module was built and tested sequentially.

4. **Integration and Testing:**

- After the platform was fully developed, a month-long integration and testing phase followed, where the entire system was tested for functionality, performance, and security.

5. **Deployment:**

- Once the system passed testing, it was deployed. Deployment took one week, followed by a post-launch review.

6. **Maintenance:**

- In the maintenance phase, any bugs or post-launch issues were addressed. However, substantial feature requests were deferred to future versions due to the rigidity of the initial design.

Key Observations from Waterfall Implementation:

• **Advantages:**

- **Clarity and Documentation:** Every phase of the project was thoroughly documented, allowing for clear guidelines and minimal confusion during implementation.
- **Predictability:** With a clear timeline and scope, the project was completed within its original budget and timeframe.

• **Challenges:**

- **Lack of Flexibility:** When new market trends emerged during development, ShopSphere couldn't easily adapt, leading to missed opportunities.
- **Customer Feedback:** Since the platform was only tested by end-users at the very end, some key customer experience issues surfaced post-launch.
- **Risk of Obsolescence:** Given the rapid pace of change in e-commerce, some features were outdated by the time the platform went live.

Approach 2: Agile Development

For their next iteration, ShopSphere adopted the Agile development model, focusing on iterative, incremental delivery with regular feedback loops from stakeholders and customers.

Agile Process for ShopSphere:

1. **Sprint Planning and Initial Backlog:**

- ShopSphere began with a high-level list of features and goals. They broke the project into several sprints, with each sprint lasting three weeks.
- During each sprint, the team focused on developing a subset of features, such as the shopping cart module or product recommendation engine.

2. Iterative Development and Continuous Feedback:

- After each sprint, the partially developed platform was demoed to stakeholders, including business teams and a small group of customers. Feedback was gathered and incorporated into the next sprint.
- The development team worked closely with marketing and customer support, ensuring that the evolving platform met real-time market demands and customer expectations.

3. Testing and Quality Assurance:

- Testing was integrated into every sprint. Developers used continuous integration tools to ensure that each new feature worked seamlessly with the existing platform.

4. Customer Involvement:

- At the end of each sprint, the team released a beta version to a group of customers, allowing them to interact with new features and provide input. This helped ShopSphere to fine-tune their platform in real-time.

5. Deployment and Maintenance:

- By the time the Agile team reached the final stages of development, the platform had already been refined through multiple rounds of testing and feedback. Deployment was smoother, and customer satisfaction was higher since many issues had been preemptively addressed.

Key Observations from Agile Implementation:

• Advantages:

- **Flexibility:** ShopSphere could pivot to accommodate new business goals or emerging technologies mid-project.
- **Customer-Centric Development:** Regular customer feedback meant that the platform evolved in line with user expectations.
- **Faster Time-to-Market:** Although the project took slightly longer overall, new features were released to market incrementally, keeping ShopSphere competitive throughout the development cycle.

• Challenges:

- **Scope Creep:** The flexibility of Agile allowed the business team to continuously add features, which occasionally led to delays in final delivery.
- **Uncertainty in Budget:** Since the project was dynamic, there were difficulties in keeping costs within the initial estimates.

Comparison of Waterfall and Agile Approaches

Aspect	Waterfall	Agile
Flexibility	Low	High
	Changes are difficult to implement	Adaptable to new requirements
Customer Involvement	Limited	High
	Customers involved post-launch	Continuous feedback from users
Risk of Delays	Low	Medium
	Strict timeline	Scope creep can introduce delays
Documentation	Extensive documentation	Minimal
		Focus is on working software
Time-to-Market	Longer overall	Incremental releases speed market entry
Cost Management	Predictable	Variable due to changing requirements

Conclusion

ShopSphere's experience with both the Waterfall and Agile models highlights the strengths and limitations of each approach within the SDLC. The Waterfall model's structured, predictable process worked well for projects with well-defined requirements but lacked the adaptability needed for evolving business landscapes. On the other hand, Agile allowed ShopSphere to remain competitive by responding to customer needs and market trends in real time, though at the cost of some uncertainty in budget and project scope.

For future development, ShopSphere decided to adopt a hybrid approach, leveraging the structured planning of Waterfall for large-scale feature development while incorporating Agile for customer-centric, iterative improvements. This hybrid strategy allowed them to combine the best of both worlds, ensuring long-term success in the dynamic e-commerce industry.

Questions

The questions and answer sheet for this case study are available on the following Google Form.

<https://forms.gle/dR3drXocUggtx8F99>
