**Các phương pháp triển khai SDLC trong việc phát triển nền tảng thương mại điện tử**  
**Tên khóa học:** CS502052 – Các khái niệm phát triển hệ thống doanh nghiệp (ESDC)  
**Giảng viên:** Phúc H. Dương

**Bối cảnh**  
Công ty thương mại điện tử ShopSphere đã lên kế hoạch nâng cấp nền tảng hiện tại của mình để đáp ứng lượng truy cập ngày càng tăng, cải thiện trải nghiệm khách hàng, và tích hợp các tính năng tiên tiến như gợi ý sử dụng AI. Công ty đã nghiên cứu hai phương pháp khác nhau để phát triển nền tảng của mình: mô hình Waterfall (thác nước) và phát triển Agile. Cả hai phương pháp đều tuân theo khuôn khổ Vòng đời phát triển phần mềm (SDLC), nhưng chúng có ảnh hưởng khác nhau đến việc thực hiện dự án, tính linh hoạt, và kết quả cuối cùng.

**Mục tiêu**  
Nghiên cứu tình huống này phân tích việc triển khai cả hai mô hình Waterfall và Agile trong dự án phát triển nền tảng thương mại điện tử của ShopSphere, làm nổi bật các điểm mạnh và điểm yếu của từng mô hình liên quan đến yêu cầu dự án, thực hiện, và kết quả cuối cùng.

**Phương pháp 1: Mô hình Waterfall**

Mô hình Waterfall tuân theo quy trình thiết kế tuần tự, trong đó mỗi giai đoạn phải được hoàn thành trước khi chuyển sang giai đoạn tiếp theo. Đối với ShopSphere, phương pháp này ban đầu thu hút bởi tính cấu trúc và thời gian hoàn thành rõ ràng.

**Các giai đoạn trong Waterfall cho ShopSphere:**

1. **Phân tích yêu cầu:**
   * Nhóm kinh doanh của ShopSphere đã dành hai tháng để thu thập yêu cầu, bao gồm các thông số kỹ thuật về tính năng, nhu cầu khách hàng, và các nâng cấp cơ sở hạ tầng kỹ thuật. Giai đoạn này rất kỹ lưỡng, đảm bảo rằng tất cả các chi tiết đã được vạch ra trước khi tiến hành.
2. **Thiết kế hệ thống:**
   * Sau khi yêu cầu được xác định, một nhóm kiến trúc sư đã thiết kế hệ thống. Giai đoạn này kéo dài thêm ba tháng. Các tài liệu thiết kế bao quát cấu trúc cơ sở dữ liệu, kiến trúc hệ thống, UI/UX, và tích hợp back-end.
3. **Triển khai:**
   * Nhóm phát triển bắt đầu xây dựng nền tảng dựa trên các tài liệu thiết kế chi tiết. Việc mã hóa kéo dài bốn tháng, khi mỗi module được xây dựng và kiểm thử theo thứ tự.
4. **Tích hợp và kiểm thử:**
   * Sau khi nền tảng được phát triển hoàn chỉnh, một giai đoạn tích hợp và kiểm thử kéo dài một tháng được tiến hành, nơi toàn bộ hệ thống được kiểm tra về tính năng, hiệu suất và bảo mật.
5. **Triển khai:**
   * Sau khi hệ thống vượt qua kiểm thử, nó được triển khai. Quá trình triển khai mất một tuần, kèm theo một đợt đánh giá sau khi ra mắt.
6. **Bảo trì:**
   * Trong giai đoạn bảo trì, bất kỳ lỗi nào hoặc vấn đề sau khi ra mắt đều được giải quyết. Tuy nhiên, các yêu cầu tính năng lớn được hoãn lại cho các phiên bản tương lai do tính cứng nhắc của thiết kế ban đầu.

**Những quan sát chính từ việc triển khai Waterfall:**

* **Ưu điểm:**
  + **Rõ ràng và có tài liệu:** Mỗi giai đoạn của dự án được ghi lại kỹ lưỡng, cung cấp các hướng dẫn rõ ràng và hạn chế sự nhầm lẫn trong quá trình triển khai.
  + **Dễ dự đoán:** Với thời gian và phạm vi rõ ràng, dự án đã hoàn thành trong ngân sách và thời gian dự kiến.
* **Thách thức:**
  + **Thiếu tính linh hoạt:** Khi các xu hướng thị trường mới xuất hiện trong quá trình phát triển, ShopSphere không thể dễ dàng thích ứng, dẫn đến việc bỏ lỡ các cơ hội.
  + **Phản hồi khách hàng:** Vì nền tảng chỉ được kiểm thử bởi người dùng cuối vào cuối cùng, một số vấn đề về trải nghiệm khách hàng chỉ được phát hiện sau khi ra mắt.
  + **Nguy cơ lỗi thời:** Với tốc độ thay đổi nhanh chóng trong lĩnh vực thương mại điện tử, một số tính năng đã lỗi thời khi nền tảng được ra mắt.

**Phương pháp 2: Phát triển Agile**

Trong lần phát triển tiếp theo, ShopSphere đã áp dụng mô hình phát triển Agile, tập trung vào việc phát triển theo từng giai đoạn và nhận phản hồi liên tục từ các bên liên quan và khách hàng.

**Quy trình Agile cho ShopSphere:**

1. **Lập kế hoạch Sprint và Backlog ban đầu:**
   * ShopSphere bắt đầu với một danh sách cấp cao các tính năng và mục tiêu. Họ chia dự án thành nhiều sprint, mỗi sprint kéo dài ba tuần. Trong mỗi sprint, nhóm tập trung phát triển một phần các tính năng, như module giỏ hàng hoặc hệ thống gợi ý sản phẩm.
2. **Phát triển lặp lại và phản hồi liên tục:**
   * Sau mỗi sprint, nền tảng được phát triển một phần sẽ được trình bày cho các bên liên quan, bao gồm nhóm kinh doanh và một nhóm nhỏ khách hàng. Phản hồi được thu thập và tích hợp vào sprint tiếp theo. Nhóm phát triển làm việc chặt chẽ với nhóm tiếp thị và hỗ trợ khách hàng để đảm bảo rằng nền tảng đang phát triển đáp ứng được các yêu cầu thực tế của thị trường và kỳ vọng của khách hàng.
3. **Kiểm thử và đảm bảo chất lượng:**
   * Kiểm thử được tích hợp vào mỗi sprint. Các nhà phát triển sử dụng các công cụ tích hợp liên tục để đảm bảo rằng mỗi tính năng mới hoạt động một cách liền mạch với nền tảng hiện có.
4. **Sự tham gia của khách hàng:**
   * Cuối mỗi sprint, nhóm phát hành một phiên bản beta cho một nhóm khách hàng, cho phép họ tương tác với các tính năng mới và cung cấp phản hồi. Điều này giúp ShopSphere tinh chỉnh nền tảng theo thời gian thực.
5. **Triển khai và bảo trì:**
   * Khi nhóm Agile đạt đến các giai đoạn cuối của phát triển, nền tảng đã được tinh chỉnh qua nhiều vòng kiểm thử và phản hồi. Việc triển khai trở nên mượt mà hơn và sự hài lòng của khách hàng cao hơn vì nhiều vấn đề đã được giải quyết trước đó.

**Những quan sát chính từ việc triển khai Agile:**

* **Ưu điểm:**
  + **Tính linh hoạt:** ShopSphere có thể điều chỉnh để đáp ứng các mục tiêu kinh doanh mới hoặc công nghệ đang nổi trong suốt quá trình dự án.
  + **Phát triển hướng tới khách hàng:** Phản hồi khách hàng thường xuyên giúp nền tảng phát triển theo đúng kỳ vọng của người dùng.
  + **Thời gian ra mắt nhanh hơn:** Mặc dù dự án kéo dài lâu hơn một chút, nhưng các tính năng mới được phát hành ra thị trường một cách liên tục, giúp ShopSphere duy trì tính cạnh tranh trong suốt chu kỳ phát triển.
* **Thách thức:**
  + **Lạm phát phạm vi:** Tính linh hoạt của Agile cho phép nhóm kinh doanh liên tục bổ sung các tính năng mới, đôi khi gây ra sự chậm trễ trong việc hoàn thành cuối cùng.
  + **Không chắc chắn về ngân sách:** Vì dự án thay đổi động, việc giữ chi phí trong ước tính ban đầu trở nên khó khăn.

**So sánh giữa các phương pháp Waterfall và Agile**

| **Khía cạnh** | **Waterfall** | **Agile** |
| --- | --- | --- |
| **Tính linh hoạt** | Thấp (khó thay đổi) | Cao (thích ứng với yêu cầu mới) |
| **Sự tham gia của khách hàng** | Hạn chế (chỉ sau khi ra mắt) | Cao (phản hồi liên tục từ người dùng) |
| **Nguy cơ trễ hạn** | Thấp (thời gian nghiêm ngặt) | Trung bình (lạm phát phạm vi có thể gây chậm trễ) |
| **Tài liệu** | Tài liệu chi tiết | Tối thiểu (tập trung vào phần mềm hoạt động) |
| **Thời gian ra mắt** | Lâu hơn tổng thể | Phát hành dần dần giúp ra mắt nhanh hơn |
| **Quản lý chi phí** | Dễ dự đoán | Biến động do yêu cầu thay đổi |

**Kết luận**  
Trải nghiệm của ShopSphere với cả hai mô hình Waterfall và Agile làm nổi bật những điểm mạnh và hạn chế của từng phương pháp trong khuôn khổ SDLC. Quy trình cấu trúc và có thể dự đoán của Waterfall hoạt động tốt cho các dự án có yêu cầu rõ ràng, nhưng thiếu tính linh hoạt cần thiết trong các bối cảnh kinh doanh thay đổi. Mặt khác, Agile giúp ShopSphere giữ được tính cạnh tranh bằng cách đáp ứng nhu cầu khách hàng và xu hướng thị trường theo thời gian thực, dù phải đối mặt với sự không chắc chắn về ngân sách và phạm vi dự án.

Trong các dự án tương lai, ShopSphere quyết định áp dụng phương pháp kết hợp, tận dụng việc lập kế hoạch cấu trúc của Waterfall cho các tính năng lớn, đồng thời kết hợp Agile cho các cải tiến lặp lại, hướng tới khách hàng. Chiến lược này cho phép họ kết hợp những ưu điểm của cả hai mô hình, đảm bảo thành công lâu dài trong ngành thương mại điện tử năng động.