BÀI TẬP BUỔI 4

Bài 1.1

**a) Dự án kết thúc khi?** (chọn nhiều)

* **Đáp án đúng**: 1, 2, 3, 4, 5  
  Giải thích: Dự án có thể kết thúc do các lý do như thiếu kinh phí, hết kinh phí trước thời hạn (kết thúc thất bại), không khả thi, quá hạn dự kiến hoặc không còn ý nghĩa để tiếp tục, hoặc hoàn thành mục tiêu đề ra và nghiệm thu thành công.

**b) Một dự án phát triển phần mềm là thành công khi?**

* **Đáp án đúng**: 4. Tất cả các phương án trên  
  Giải thích: Một dự án phần mềm được coi là thành công khi sản phẩm đáp ứng yêu cầu chất lượng, hoàn thành trong thời gian dự kiến và không vượt quá kinh phí dự kiến.

**c) ... là một yếu tố không biết trước mà khi nó xảy ra thì có thể ảnh hưởng tiêu cực hoặc tích cực đến việc hoàn thành các mục tiêu của dự án.**

* **Đáp án đúng**: 1. Rủi ro dự án  
  Giải thích: Rủi ro dự án là yếu tố không biết trước có thể ảnh hưởng đến kết quả dự án, bao gồm các ảnh hưởng tiêu cực hoặc tích cực đến việc hoàn thành các mục tiêu của dự án.

Bài 1.2

**a) Đối với một người tham gia dự án mà không biết về toàn bộ chức năng của dự án có hợp lý không? Vì sao?**

Có thể coi là hợp lý khi một cá nhân tham gia vào một phần của dự án mà không biết hết toàn bộ chức năng của dự án. Nguyên nhân là vì trong các hệ thống phần mềm lớn và phức tạp, công việc thường được chia nhỏ thành các module hoặc thành phần, và các thành viên được phân công phát triển những phần cụ thể này. Điều này giúp chuyên môn hóa, cho phép các cá nhân tập trung vào phần việc của mình một cách hiệu quả mà không cần phải biết rõ về mọi chi tiết của dự án. Bên cạnh đó, các thành viên trong nhóm vẫn được hỗ trợ với tài liệu, họp định kỳ, và các công cụ quản lý để hiểu các yêu cầu và mục tiêu tổng thể, đảm bảo rằng công việc của họ phù hợp với mục tiêu chung của dự án.

**b) Nếu không áp dụng các mô hình vòng đời phần mềm thì có phát triển được phần mềm không? Tại sao?**

Phát triển phần mềm mà không áp dụng các mô hình vòng đời sẽ gặp nhiều khó khăn và rủi ro hơn, nhưng không hoàn toàn là không thể. Các mô hình vòng đời phần mềm (như Waterfall, Agile) cung cấp các quy trình, cấu trúc, và hướng dẫn rõ ràng, giúp đảm bảo rằng mọi bước trong quá trình phát triển được thực hiện hiệu quả, từ việc thu thập yêu cầu, thiết kế, phát triển đến kiểm thử và bảo trì. Nếu không có một mô hình cụ thể, dự án dễ bị mất định hướng, khó quản lý tiến độ và chất lượng, và có thể dẫn đến lãng phí tài nguyên do sự thiếu tổ chức. Vì vậy, việc áp dụng các mô hình vòng đời là rất quan trọng để đảm bảo hiệu quả và thành công của dự án.

**c) Trong phương pháp Agile, việc luôn có đại diện của khách hàng trong nhóm phát triển thì có ưu điểm gì?**

Việc luôn có đại diện của khách hàng trong nhóm phát triển theo phương pháp Agile mang lại nhiều lợi ích:

1. **Đảm bảo tính chính xác của yêu cầu**: Đại diện khách hàng có thể cung cấp phản hồi nhanh chóng và điều chỉnh yêu cầu nếu cần, giúp nhóm phát triển hiểu rõ và đáp ứng đúng nhu cầu của người dùng.
2. **Giảm thiểu rủi ro**: Sự có mặt của khách hàng giúp phát hiện sớm các sai lệch hoặc vấn đề, giảm thiểu rủi ro phải làm lại sản phẩm sau khi hoàn thành.
3. **Tăng tính linh hoạt**: Agile coi trọng khả năng thích ứng với thay đổi. Khi có khách hàng cùng tham gia, các thay đổi về yêu cầu có thể được trao đổi và thực hiện nhanh chóng.
4. **Cải thiện chất lượng sản phẩm**: Thông qua việc khách hàng thường xuyên tham gia vào các buổi kiểm tra và đánh giá, nhóm phát triển có thể tạo ra sản phẩm đáp ứng sát với kỳ vọng của khách hàng hơn.

PHẦN II

**a) Biểu đồ WBS cho dự án phát triển phần mềm**

Dưới đây là một ví dụ về WBS cho dự án phát triển phần mềm với cấu trúc phân rã theo các pha trong quy trình phát triển phần mềm:

**Dự án phát triển phần mềm**

1. **Khởi tạo dự án**
   * 1.1 Xác định yêu cầu dự án
   * 1.2 Phân tích rủi ro ban đầu
   * 1.3 Phân bổ tài nguyên và lập kế hoạch sơ bộ
2. **Thu thập và phân tích yêu cầu**
   * 2.1 Gặp gỡ khách hàng để lấy yêu cầu
   * 2.2 Phân tích yêu cầu hệ thống
   * 2.3 Tạo tài liệu yêu cầu
3. **Thiết kế hệ thống**
   * 3.1 Thiết kế kiến trúc hệ thống
   * 3.2 Thiết kế cơ sở dữ liệu
   * 3.3 Thiết kế giao diện người dùng
4. **Phát triển và lập trình**
   * 4.1 Lập trình back-end
   * 4.2 Lập trình front-end
   * 4.3 Kết nối API và dịch vụ
   * 4.4 Tích hợp hệ thống
5. **Kiểm thử**
   * 5.1 Kiểm thử đơn vị
   * 5.2 Kiểm thử tích hợp
   * 5.3 Kiểm thử hệ thống
   * 5.4 Kiểm thử chấp nhận người dùng
6. **Triển khai**
   * 6.1 Chuẩn bị môi trường triển khai
   * 6.2 Đào tạo người dùng cuối
   * 6.3 Triển khai sản phẩm
7. **Bảo trì và hỗ trợ**
   * 7.1 Giám sát hiệu suất
   * 7.2 Xử lý lỗi và cập nhật phần mềm
   * 7.3 Hỗ trợ người dùng và cung cấp tài liệu

Bản kế hoạch đơn giản dựa trên WBS:

|  |  |  |
| --- | --- | --- |
| Công việc | Thời gian (số giờ làm việc) | Số người |
| 1. Khởi tạo dự án | 20 | 2 |
| 1.1 Xác định yêu cầu dự án | 5 | 1 |
| 1.2 Phân tích rủi ro ban đầu | 5 | 1 |
| 1.3 Phân bổ tài nguyên | 10 | 1 |
| 2. Thu thập và phân tích yêu cầu | 60 | 3 |
| 2.1 Gặp gỡ khách hàng | 20 | 1 |
| 2.2 Phân tích yêu cầu | 20 | 2 |
| 2.3 Tạo tài liệu yêu cầu | 20 | 2 |
| 3. Thiết kế hệ thống | 100 | 4 |
| 3.1 Thiết kế kiến trúc | 40 | 2 |
| 3.2 Thiết kế cơ sở dữ liệu | 30 | 1 |
| 3.3 Thiết kế giao diện | 30 | 1 |

b) Bảng quản lý rủi ro trong dự án phát triển phần mềm

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Công việc/hoạt động | Mối nguy | Rủi ro | Mức độ | Chiến lược | Biện pháp |
| Khởi tạo dự án | Thiếu tài nguyên nhân sự | Kéo dài tiến độ | Trung bình | Giảm thiểu | Tuyển dụng bổ xung |
| Phân tích yêu cầu | Yêu cầu không rõ | Phát triển sai chức năng | Cao | Giảm thiểu | Gặp khách hàng thường xuyên |
| Thiết kế hệ thống | Thiết kế sai cơ sở dữ liệu | Phải làm lại hệ thống | Cao | Loại bỏ | Tạo tài liệu kỹ lưỡng, kiểm tra lại |
| Phát triển | Lỗi code | Sản phẩm không đạt yêu cầu | Cao | Giảm thiểu | Kiểm thử thường xuyên |
| Triển khai | Môi trường không thích hợp | Không thể triển khai | Trung bình | Giảm thiểu | Chuẩn bị môi trường từ sớm |
| Bảo trì và hỗ trợ | Không đủ tài liệu hướng dẫn | Người dùng gặp khó khăn | Thấp | Giảm thiểu | Chuẩn bị tài liệu kỹ lưỡng |