**Bài tập tuần 04**

**Quản lý dự án phần mềm &**

**Lập trình với giao diện đồ hoạ người dùng (GUI)**

**Phần I:**

**Bài 1.1**

1. Dự án kết thúc khi? (chọn nhiều)
2. Thiếu kinh phí
3. Hết kinh phí trước thời hạn (Kết thúc thất bại)
4. **Không khả thi**
5. Quá  hạn dự kiến (có làm tiếp cũng không có ý nghĩa gì)
6. **Hoàn thành mục tiêu đề ra, nghiệm thu (kết thúc tốt đẹp) trước thời hạn**
7. Người quản lý kém
8. Một dự án phát triển phầm mềm là thành công khi?
9. Sản phẩm đáp ứng yêu cầu chất lượng
10. Không vượt quá kinh phí dự kiến
11. Hoàn thành trong thời gian dự kiến
12. **Tất cả các phương án trên**
13. … 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.
14. **Rủi ro dự án**
15. Lập kế hoạch dự án
16. Quản lý nhân sự
17. Yêu cầu người dùng

**Bài 1.2**

1. Chúng ta thấy rằng các hệ thống phần mềm lớn, phức tạp thường được phát triển bởi rất nhiều cá nhân, rất ít người có được bức tranh toàn cảnh về toàn bộ dự án. Vậy, đối với một người làm công, tham gia vào một dự án mà không biết về toàn bộ chức năng của dự án đó thì có hợp lý không? Vì sao?

Đối với 1 người làm công , tham gia vào một dự án mà không biết về toàn bộ chức năng của dự án đó thì cũng hoàn toàn hợp lí . bởi lẽ : mỗi người phụ trách một mảnh của dự án được phân công , khi đó người làm công không nhất thiết phải biết toàn bộ chức năng của dự án , mỗi người được phân công việc cần hiểu rõ những nhiệm vụ và công việc được giao và hoàn thành nó để góp phần tạo dựng lên một hệ thống phần mềm lớn từ nhiều bộ phận , cá nhân hợp sức .

1. Nếu không áp dụng các mô hình vòng đời phần mềm thì có phát triển được phần mềm không? Tại sao?

Việc áp dụng các mô hình vòng đời phần mềm giúp phát triển phần mềm một cách tốt hơn. Việc nắm rõ các mô hình vòng đời phát triển và kiểm thử phần mềm giúp cho bạn có định hướng đúng đắn hơn và xây dựng hoạt động kiểm thử hiệu quả và phù hợp hơn từ đó giải quyết và lường trước được những phát

sinhh và rủi ro hơn là khi không áp dụng các mô hình vòng đời . mỗi mô hình vòng đời phần mềm đều có những ưu điểm và nhược điểm.

1. Trong phương pháp Agile, việc luôn có đại diện của khác hàng trong nhóm phát triển thì có ưu điểm gì?

Ưu điểm của việc có đại diện của khách hàng trong nhóm phát triển:

Có thể truyền đạt cái nhìn tổng thể , sứ mệnh của sản phẩm của nhóm đang xây dựng , Mỗi thành viên của nhóm phát triển là một phần không thể thiếu trong việc đưa ra các giải pháp và dự kiến sẽ thực hiện một sản phẩm từ khi bắt đầu đến khi hoàn thành . từ đó có trách nhiệm quản lí sản phẩm và hoàn thành phần tăng trưởng của công việc .

**Phần II: Xây dựng biểu đồ WBS**

**a) Yêu cầu: Nhóm sinh viên thảo luận và xây dựng WBS đối với dự án phát triển phần mềm trong bài tập môn học. Có thể phân tích theo các chức năng nghiệp vụ hoặc theo các pha trong quá trình phát triển.**

|  |  |  |
| --- | --- | --- |
| Công việc | Thời gian (số giờ làm việc) | Số người |
| **1.Phân tích yêu cầu phần mềm quản lý thông tin Covid-19**  1.1 Xác định các yêu cầu  1.2 Thay đổi yêu cầu phần mềm  1.3 Thống nhất yêu cầu phần mềm với khách hàng | 2-5 ngày | 2-4 người |
| **2.Thiết kế phần mềm** 2.1 Xây dựng biểu đồ đặc tả | 5-10 ngày | 2-6 người |
| **3. Code tính năng, code hệ thống** | 5-30 ngày | 2-8 người |
| **4. Code giao diện**  4.1 Xác định giao diện  4.2 Thiết kế giao diện  4.3 Hoàn thành biểu mẫu khai báo y tế | 3-10 ngày | 2-3 người |
| **5. Tạo lập Cơ sở dữ liệu** | 3-10 ngày | 2-3 người |
| **6. Kiểm thử phần mềm** | 3-10 ngày | 2-3 người |
| **7.Triển khai phần mềm** | 3-10 ngày | 2-5 người |
| **8. Bảo trì phần mềm** | 3-10 ngày | 2-8 người |

**b) Yêu cầu: Xây dựng một bảng quản lý đơn giản các rủi ro với dự án phát triển phần mềm trong bài tập môn học.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Công việc / Hoạt động | Xác định rủi ro | | | Quản lý rủi ro | |
| **Mối nguy** | **Rủi ro** | **Mức độ** | **Chiến lược** | **Biện pháp** |
| **Phân tích yêu cầu phần mềm** | Không xác định rõ ràng yêu cầu | Yêu cầu không hoàn chỉnh | Cao | Loại bỏ rủi ro | Đặc tả rõ yêu cầu phần mềm, mô hình hoá yêu cầu |
| **Thiết kế xây dựng phần mềm** | Phần mềm thiết kế không rõ ràng, khó hiểu | Khó khăn khi code sản phẩm phần mềm | Trung bình | Giảm thiểu rủi ro | Thiết kế xây dựng phần mềm rõ ràng, vẽ biểu đồ,  tuân thủ các nguyên tắc cơ bản như KISS, DRY, SOLID trong thiết kế phần mềm |
| **Code giao diện** | Không xác định đầy đủ yêu cầu cho giao diện | Giao diện chưa đầy đủ, chưa đáp ứng được nhu cầu khách hàng | Cao | Loại bỏ rủi ro | Phân tích, xác định rõ ràng |
| **Kiểm thử phần mềm** | Không phát hiện ra đầy đủ sai sót | Phần mềm hoạt động phát sinh bị lỗi | Cao | Loại bỏ rủi ro | Tuân thủ 7 nguyên tắc trong kiểm thử |