Họ và tên: Phạm Mạnh Cường   
MSSV: N19DCAT0   
 Nhập môn công nghệ phần mềm.

Tuần 3

**Part 1: Câu hỏi trắc nghiệm**

**Câu hỏi 1: Pha nào trong mô hình lý thuyết vòng đời phát triển phần mềm chịu trách nhiệm chuyển đổi yêu cầu thành đặc tả kỹ thuật?**

* **Đáp án đúng: C. Pha phân tích**
* **Giải thích:** Pha phân tích là giai đoạn quan trọng trong vòng đời phát triển phần mềm, nơi các yêu cầu của khách hàng hoặc người dùng được xem xét và chuyển đổi thành tài liệu đặc tả kỹ thuật. Đặc tả kỹ thuật này giúp lập trình viên và các bên liên quan hiểu rõ hệ thống cần phát triển.

**Câu hỏi 2: Mô hình vòng đời nào phát triển phần mềm bằng cách tạo các phiên bản nhỏ và tăng dần tính năng?**

* **Đáp án đúng: B. Mô hình lặp và tăng trưởng**
* **Giải thích:** Mô hình **lặp và tăng trưởng (Incremental and Iterative Model)** cho phép phát triển phần mềm theo từng phiên bản nhỏ, mỗi phiên bản cải thiện hoặc bổ sung tính năng mới. Điều này giúp phần mềm linh hoạt hơn, có thể thích nghi với các thay đổi nhanh chóng trong yêu cầu của người dùng.

**Câu hỏi 3: Pha bảo trì trong vòng đời phát triển phần mềm bao gồm hoạt động nào?**

* **Đáp án đúng: B. Sửa lỗi và cập nhật tính năng mới**
* **Giải thích:** Giai đoạn bảo trì trong vòng đời phần mềm bao gồm việc **sửa lỗi (bug fixing)**, **tối ưu hiệu suất**, **nâng cấp tính năng** và **cập nhật bảo mật**. Viết mã nguồn (A) không thuộc giai đoạn bảo trì, mà thuộc giai đoạn phát triển. Gỡ bỏ phần mềm (C) liên quan đến giai đoạn ngừng hỗ trợ chứ không phải bảo trì.

**Câu hỏi 4: Mô hình thác nước phù hợp nhất với loại dự án nào?**

* **Đáp án đúng: B. Dự án có yêu cầu rõ ràng và ít thay đổi**
* **Giải thích:** Mô hình **thác nước (Waterfall Model)** là một phương pháp phát triển phần mềm tuyến tính, trong đó mỗi pha phải hoàn thành trước khi chuyển sang pha tiếp theo. Do đó, nó phù hợp với các dự án **có yêu cầu rõ ràng ngay từ đầu và ít thay đổi**. Nếu yêu cầu thay đổi nhiều (C, D), mô hình thác nước sẽ không linh hoạt bằng các mô hình khác như Agile hay mô hình xoắn ốc.

**Câu hỏi 5: Trong mô hình xoắn ốc, mỗi vòng xoắn tương ứng với:**

* **Đáp án đúng: B. Một chu kỳ lặp của toàn bộ quy trình phát triển**
* **Giải thích:** Mô hình **xoắn ốc (Spiral Model)** kết hợp các yếu tố của mô hình thác nước và mô hình lặp. Mỗi vòng xoắn tương ứng với một chu kỳ phát triển, bao gồm **phân tích rủi ro, thiết kế, triển khai, kiểm thử và đánh giá**. Việc lặp lại liên tục giúp cải thiện sản phẩm theo từng giai đoạn, đảm bảo phần mềm đáp ứng tốt nhất nhu cầu thực tế.

**Câu hỏi 6: Điểm yếu lớn nhất của mô hình xây và sửa là gì?**

* **Đáp án đúng: C. Khó kiểm soát chất lượng**
* **Giải thích:** Mô hình **xây và sửa (Build and Fix)** là phương pháp phát triển phần mềm đơn giản nhất, trong đó phần mềm được xây dựng và chỉnh sửa ngay khi có lỗi mà không có kế hoạch rõ ràng. Điều này dẫn đến:
  + Khó kiểm soát chất lượng sản phẩm do thiếu quy trình kiểm thử bài bản.
  + Code dễ bị lỗi và khó bảo trì.
  + Không phù hợp với các dự án lớn, yêu cầu độ tin cậy cao.

**Câu hỏi 7: Mô hình nào tập trung vào việc tạo các nguyên mẫu nhanh để thu thập phản hồi từ khách hàng?**

* **Đáp án đúng: B. Mô hình bản mẫu nhanh (Prototyping Model)**
* **Giải thích:** Mô hình **bản mẫu nhanh** cho phép phát triển nhanh một phiên bản thử nghiệm của phần mềm để khách hàng đánh giá và phản hồi. Điều này giúp giảm rủi ro và đảm bảo rằng sản phẩm cuối cùng đáp ứng nhu cầu thực tế của người dùng.

**Câu hỏi 8: Pha nào kết thúc vòng đời phát triển phần mềm?**

* **Đáp án đúng: C. Pha giải thể**
* **Giải thích:** Vòng đời phát triển phần mềm không chỉ dừng lại ở **triển khai và bảo trì**, mà còn bao gồm **giai đoạn giải thể** khi phần mềm không còn được sử dụng. Trong pha này, hệ thống có thể bị gỡ bỏ, dữ liệu được sao lưu, và tài nguyên phần cứng có thể được tái sử dụng cho mục đích khác.

**Câu hỏi 9: Điểm khác biệt chính giữa mô hình lặp và tăng trưởng với mô hình thác nước là gì?**

* **Đáp án đúng: B. Mô hình lặp và tăng trưởng phát triển theo từng đợt nhỏ**
* **Giải thích:**
  + **Mô hình thác nước (Waterfall Model)** có tính tuần tự, từng giai đoạn phải hoàn thành trước khi chuyển sang giai đoạn tiếp theo, ít linh hoạt khi yêu cầu thay đổi.
  + **Mô hình lặp và tăng trưởng (Incremental and Iterative Model)** phát triển phần mềm theo từng đợt nhỏ, mỗi đợt hoàn thiện thêm một phần tính năng, giúp thích nghi nhanh với thay đổi của khách hàng.

**Câu hỏi 10: Mô hình nào có khả năng thích nghi tốt nhất với sự thay đổi của yêu cầu khách hàng?**

* **Đáp án đúng: D. Mô hình tiến trình linh hoạt (Agile Model)**
* **Giải thích:**
  + **Mô hình Agile** được thiết kế để thích nghi với sự thay đổi nhanh chóng của yêu cầu khách hàng.
  + Các nhóm phát triển phần mềm làm việc theo **các vòng lặp ngắn (iterations)** và có thể điều chỉnh sản phẩm sau mỗi vòng để phù hợp với yêu cầu thực tế.
  + So với **mô hình thác nước** (cứng nhắc), **xoắn ốc** (quản lý rủi ro) hay **xây và sửa** (không có kế hoạch cụ thể), Agile mang lại sự linh hoạt cao nhất.

**Part 2: Câu hỏi trả lời ngắn**

**1. Pha lấy yêu cầu là gì và có vai trò gì trong vòng đời phát triển phần mềm?**

* **Trả lời:** Pha lấy yêu cầu là giai đoạn đầu tiên trong vòng đời phát triển phần mềm, nơi nhóm phát triển thu thập và phân tích các yêu cầu từ khách hàng hoặc người dùng.
* **Vai trò:**
  + Xác định chính xác mong muốn của khách hàng.
  + Làm cơ sở cho việc thiết kế và lập kế hoạch phát triển phần mềm.

**2. Mô hình thác nước hoạt động như thế nào?**

* **Trả lời:** Mô hình **thác nước (Waterfall)** là một phương pháp phát triển phần mềm tuyến tính, trong đó mỗi pha phải hoàn thành trước khi chuyển sang pha tiếp theo.
* **Các bước:**
  1. Lấy yêu cầu
  2. Thiết kế
  3. Lập trình
  4. Kiểm thử
  5. Triển khai
  6. Bảo trì

**3. Mô hình lặp và tăng trưởng khác gì so với mô hình thác nước?**

* **Trả lời:**
  + **Mô hình thác nước** là tuyến tính, không thể quay lại giai đoạn trước.
  + **Mô hình lặp và tăng trưởng** phát triển phần mềm theo từng **đợt nhỏ**, có thể cập nhật và cải tiến dựa trên phản hồi người dùng.

**4. Mục tiêu của pha bảo trì là gì?**

* **Trả lời:** Pha bảo trì nhằm **sửa lỗi, tối ưu hiệu suất, cập nhật tính năng mới** hoặc đảm bảo phần mềm hoạt động ổn định trong môi trường thay đổi.

**5. Mô hình xây và sửa có nhược điểm gì?**

* **Trả lời:**
  + Thiếu kế hoạch rõ ràng, dễ phát sinh lỗi.
  + Chi phí bảo trì cao do không có quy trình kiểm thử chặt chẽ.
  + Phù hợp với dự án nhỏ nhưng khó mở rộng.

**6. Mô hình bản mẫu nhanh là gì?**

* **Trả lời:** Mô hình **bản mẫu nhanh (Prototyping Model)** tạo một phiên bản thử nghiệm nhanh chóng để lấy phản hồi từ khách hàng trước khi phát triển bản chính thức.
* **Ví dụ:** Khi thiết kế giao diện mới cho Shopee, nhóm UI/UX tạo bản **Prototype** để khách hàng đánh giá **trải nghiệm mua hàng** trước khi triển khai thực tế.

**7. Pha giải thể là gì?**

* **Trả lời:** Pha giải thể là giai đoạn **kết thúc vòng đời của phần mềm**, có thể bao gồm **gỡ bỏ hệ thống, chuyển dữ liệu sang hệ thống mới hoặc đóng dự án**.

**8. Mô hình xoắn ốc là gì?**

* **Trả lời:** Mô hình **xoắn ốc (Spiral Model)** kết hợp mô hình thác nước và mô hình lặp, tập trung vào **quản lý rủi ro**. Mỗi vòng xoắn bao gồm:
  1. Phân tích rủi ro
  2. Thiết kế
  3. Lập trình
  4. Kiểm thử

**9. Tại sao mô hình tiến trình linh hoạt được đánh giá cao?**

* **Trả lời:** Mô hình **tiến trình linh hoạt (Agile)** được đánh giá cao vì:
  + **Phù hợp với dự án thay đổi liên tục.**
  + **Tăng tốc độ phát triển** do làm việc theo vòng lặp ngắn.
  + **Tạo ra sản phẩm sớm** và có thể cải tiến liên tục.

**10. Điểm khác biệt chính giữa mô hình mã nguồn mở và các mô hình khác là gì?**

* **Trả lời:** Mô hình mã nguồn mở khác với các mô hình khác ở các điểm:
  + **Cộng đồng phát triển** thay vì một nhóm nội bộ.
  + **Mã nguồn công khai**, ai cũng có thể đóng góp và sử dụng.
  + **Cập nhật liên tục**, không phụ thuộc vào một công ty duy nhất.

Part 3: Câu hỏi thảo luận nhóm:

**5. Phân tích vai trò của quản lý rủi ro trong mô hình xoắn ốc**

1. Xuyên suốt toàn bộ quy trình phát triển
   * Quản lý rủi ro là trọng tâm của mô hình xoắn ốc, được thực hiện trong mỗi vòng lặp (iteration).
2. Phát hiện và đánh giá rủi ro sớm
   * Mỗi giai đoạn của mô hình đều bắt đầu bằng việc xác định các rủi ro tiềm ẩn (về kỹ thuật, tài chính, tiến độ, yêu cầu)
   * Đánh giá mức độ nghiêm trọng và khả năng xảy ra của từng rủi ro.
3. Lập kế hoạch ứng phó rủi ro:
   * Xây dựng các phương án phòng ngừa (mitigation) hoặc giải pháp khắc phục (contingency) cho từng rủi ro cụ thể.
4. Giám sát và kiểm soát rủi ro
   * Liên tục theo dõi các rủi ro trong quá trình phát triển.
   * Cập nhật kế hoạch rủi ro khi có sự thay đổi yêu cầu hoặc tình huống mới phát sinh.
5. Đảm bảo tính linh hoạt và thích ứng
   * Khi rủi ro trở thành hiện thực, đội dự án có thể thay đổi hướng đi hoặc điều chỉnh kế hoạch để giảm thiểu tác động tiêu cực..

**6. Khi nào nên sử dụng mô hình thác nước thay vì mô hình tiến trình linh hoạt?**

Nên sử dụng mô hình thác nước (Waterfall) khi:

1. Yêu cầu dự án rõ ràng và ổn định
   * Khách hàng cung cấp đầy đủ và chi tiết các yêu cầu ngay từ đầu
   * Yêu cầu ít có khả năng thay đổi trong suốt vòng đời dự án.
2. Dự án có quy mô nhỏ hoặc ngắn hạn:
   * Mô hình thác nước phù hợp cho các dự án đơn giản, ít phức tạp.
   * Thời gian thực hiện ngắn giúp dễ dàng kiểm soát tiến độ.
3. Đội ngũ phát triển có ít kinh nghiệm với mô hình linh hoạt:
   * Khi nhóm phát triển quen thuộc với quy trình tuần tự và khó thích ứng với các mô hình Agile hoặc Scrum.
4. Ngân sách và tiến độ dự án cố định:
   * Khó điều chỉnh ngân sách hoặc thời gian, cần một quy trình phát triển có thể dự đoán trước.
5. Dự án có tính pháp lý hoặc yêu cầu tài liệu hóa cao
   * Các ngành như y tế, tài chính, hoặc chính phủ cần tài liệu chi tiết và quy trình kiểm tra nghiêm ngặt.
6. Khách hàng không tham gia thường xuyên trong quá trình phát triển:
   * Khi không cần hoặc không thể nhận phản hồi liên tục từ khách hàng.

**Part 4: Câu hỏi tình huống**

**5. Một công ty nhỏ muốn áp dụng mô hình bản mẫu nhanh nhưng gặp khó khăn do thiếu nguồn lực. Hãy đề xuất giải pháp ?**

1. Ưu tiên hóa các yêu cầu quan trọng:
   * Tập trung phát triển các tính năng cốt lõi trước, tránh lãng phí nguồn lực vào các tính năng phụ
   * Sử dụng phương pháp **MVP (Minimum Viable Product)** để tạo bản mẫu đơn giản nhưng vẫn thể hiện được giá trị chính.
2. Sử dụng các công cụ tạo bản mẫu nhanh:
   * Tận dụng các công cụ miễn phí hoặc chi phí thấp như Figma, Adobe XD, Sketch cho thiết kế giao diện.
   * Sử dụng các framework có sẵn như Bootstrap, React, hoặc các nền tảng no-code/low-code (ví dụ: Bubble, Webflow) để tiết kiệm thời gian phát triển.
3. Tái sử dụng mã nguồn và thư viện có sẵn:
   * Sử dụng các thư viện mã nguồn mở (open-source) và các plugin giúp giảm thời gian phát triển bản mẫu.
4. Thuê ngoài hoặc hợp tác với freelancer:
   * Nếu nguồn lực nội bộ không đủ, có thể thuê các chuyên gia làm việc ngắn hạn để hoàn thành các phần quan trọng của bản mẫu.
5. Linh hoạt trong việc xây dựng bản mẫu:
   * Xây dựng bản mẫu ở mức độ đơn giản hóa tối đa (ví dụ: chỉ làm các mockup hoặc prototype tương tác) thay vì phát triển đầy đủ chức năng.
6. Tận dụng phản hồi sớm từ khách hàng:
   * Sớm đưa bản mẫu ra thử nghiệm với khách hàng để thu thập phản hồi, từ đó điều chỉnh kịp thời và tránh lãng phí nguồn lực.

**6. Trong dự án phần mềm thương mại điện tử, khách hàng liên tục yêu cầu thay đổi giao diện. Mô hình nào sẽ phù hợp nhất?**

Trong trường hợp này, **mô hình tiến trình linh hoạt (Agile) sẽ là phù hợp nhất.**

**Lý do:**

1. **Linh hoạt với thay đổi:**

* Agile cho phép thay đổi yêu cầu liên tục ngay cả khi dự án đã tiến xa, phù hợp với tình huống khách hàng muốn điều chỉnh giao diện thường xuyên.

1. **Phát triển theo từng giai đoạn (Iteration):**

* Scrum (một framework của Agile) chia dự án thành các Sprint ngắn (2-4 tuần), sau mỗi Sprint đều có bản demo sản phẩm để khách hàng xem và phản hồi.

1. **Tăng cường tương tác với khách hàng:**

* Agile khuyến khích sự tham gia của khách hàng trong suốt quá trình phát triển, giúp đội phát triển nắm bắt và điều chỉnh giao diện kịp thời.

1. **Ưu tiên tính năng quan trọng:**

* Dựa trên phản hồi khách hàng, nhóm phát triển có thể ưu tiên cải thiện các yếu tố giao diện mà khách hàng quan tâm nhất.

1. **Giảm rủi ro và tối ưu nguồn lực:**

* Bằng cách thử nghiệm và điều chỉnh giao diện liên tục, Agile giúp tránh việc xây dựng sai hướng, tiết kiệm thời gian và chi phí.