Sinh viên thực hiện:

* Lê Huyền Linh – 20187178
* Nguyễn Minh Đức – 20187162
* Lê Đức Thiện - 20187200

**Bài thực hành tuần 02**

**Bài 1.1**

a) Mô hình bản mẫu (prototyping model) của phát triển phần mềm là …

1. Một cách tiếp cận hợp lý khi yêu cầu được định nghĩa rõ ràng

2. Một cách tiếp cận hữu ích khi khách hàng không thể định nghĩa yêu cầu rõ ràng

3. Cách tiếp cận tốt nhất cho những dự án có đội phát triển lớn

4. Tất cả các phương án trên đều sai

b) Bước đầu tiên trong vòng đời phát triển phần mềm (Software Development Life

Cycle) là?

1. Xác định các nhu cầu và ràng buộc

2. Viết phần mềm

3. Vận hành hệ thống để loại bỏ các khiếm khuyết

4. Nâng cao sản phẩm sau khi đã triển khai

c) Tình huống nào không phù hợp để có thể áp dụng mô hình thác nước? (chọn nhiều)

1. Khó khăn trong việc bổ sung yêu cầu mới trong các pha sau của tiến trình

2. Các yêu cầu được xác định rõ ràng, đầy đủ ngay từ đầu

3. Khách hàng muốn có sản phẩm vận hành được sớm

4. Khó thu thập đầy đủ yêu cầu ngay ở giai đoạn đầu của dự án

**Bài 1.2**

Hãy giải ô chữ dưới đây với các gợi ý kèm theo?

* 1: software life cycle
* 2: Incremental model
* 3: Software process
* 4: Prototyping
* 5: Spiral model
* 6: RAD
* 7: Waterfall
* 8: Testing
* 9: Maintenance
* 10: Componant based

**Bài 1.3**

Hãy so sánh các mô hình phát triển phần mềm: Thác nước, Chế thử, Gia tăng, Xoắn ốc,

Phát triển dựa trên thành phần

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Đặc điểm chính | Ưu điểm | Nhược điểm | Tình huống áp dụng phù hợp |
| Mô hình thác nước | +Mô hình phát triển phần mềm đầu tiên được sử dụng.  +Áp dụng tuần tự các giai đoạn của phát triển phần mềm.  +Đầu ra của giai đoạn trước là đầu vào của giai đoạn sau.  +Giai đoạn sau chỉ được thực hiện khi giai đoạn trước đã kết thúc. | +Dễ sử dụng, dễ tiếp cận, dễ quản lí.  +Sản phẩm theo các giai đoạn được xác định rõ ràng.  +Xác nhận ở từng giai đoạn đảm bảo phát hiện sớm các lỗi. | +Ít linh hoạt, phạm vi điều chỉnh hạn chế.  +Rất khó để đo lường sự phát triển trong từng giai đoạn.  +Mô hình không thích hợp với những dự án dài,hay những dự án phức tạp, có nhiều thay đổi về yêu cầu trong vòng đời phát triển.  +Khó quay lại khi giai đoạn nào đó đã kết thúc. | +Các dự án ngắn hạn.  +Các dự án có ít thay đổi về yêu cầu và không có những yêu cầu không rõ ràng. |
| Mô hình chế thử (Prototype model) | +Mô hình chế thử là 1 mô hình phát triển phần mềm được phát triển dựa trên các yêu cầu hệ thống. Dựa vào bản prototype mà khách hàng có cái nhìn tổng quan về hệ thống thực tế.  +Prototype là một ý tưởng hay cho các hệ thống phức tạp và lớn mà không có quy trình thủ công để giúp xác định yêu cầu.  +Prototype thường không phải là hệ thống hoàn chỉnh và nhiều chi tiết không được xây dựng. Mục tiêu là cung cấp một hệ thống với chức năng tổng thể. | +Người dùng tích cực tham gia vào quá trình phát triển phần mềm và từ đó hiểu rõ về hệ thống đang phát triển.  +Lỗi có thể được phát hiện sớm và được fix sớm hơn.  +Người dùng có thể nhìn thấy những điểm bất cập và đưa ra giải pháp để giải quyết sớm hơn.  +Tránh tình trạng bị thiếu mấy chức năng của hệ thống.  +Những chức năng phức tạp sẽ được xác định và giải thích sớm. | +Trên thực tế bản prototype có thể làm tăng sự phức tạp của hệ thống vì phạm vi của hệ thống có thể mở rộng hơn các kế hoạch ban đầu.  +Bản mẫu này sẽ được chính khách hàng nhận xét đánh giá, nếu bản prototype mà khách hàng không hài lòng phải làm lại gây tốn thời gian và công sức. | +Mô hình prototype nên được sử dụng khi hệ thống cần có nhiều tương tác với người dùng cuối.  +Thông thường, các hệ thống trực tuyến, giao diện web có lượng tương tác cao với người dùng cuối phù hợp nhất với mô hình Prototype. |
| Mô hình gia tăng | +Với phương pháp chia nhỏ thành từng bản khác nhau, thì từ 1 chu kì lớn sẽ được chia thành các chu kì nhỏ ứng với từng bản, và do đó ta được 1 đa chu kì phát triển.  +Một chu kì nhỏ ứng với một bản phân chia gọi là module đơn giản hơn và dễ dàng quản lí hơn mỗi module cũng được xây dựng theo từng bước như là phân tích, đọc yêu cầu dự án, viết thiết kế, tiến hành coding và test. | +Có thể sớm tạo ra các nguyên mẫu của sản phẩm trong vòng đời phát triển của nó.  +Độ linh hoạt cao hơn và khi thay đổi sẽ cần chi phí ít hơn, vì những thay đổi thuộc về module nào thì module đó sẽ thay đổi mà các module khác không bị ảnh hưởng.  +Giảm chi phí cho lần đầu giao sản phẩm.  +Dễ dàng quản lí các rủi ro có thể phát sinh. | +Cần phải có khả năng thiết kế tốt, để có thể hiểu rõ được yêu cầu và biết cách phân chia ra cho hợp lý.  +Chi phí để phát triển theo phương pháp này là rất cao. | +Áp dụng mô hình này khi yêu cầu của dự án.  +Khi sớm cần có một nguyên mẫu phần mềm quảng bá, giới thiệu, hoặc thử nghiệm.  +Sử dụng mô hình này khi một công nghệ mới được áp dụng.  +Tài nguyên và kĩ năng chuyên môn luôn sẵn sàng.  +Khi có một tính năng hay các mục tiêu có nguy cơ lỗi cao. |
| Mô hình xoắn ốc | +Mô hình xoắn ốc (Spiral-Model) là mô hình có sự kết hợp giữa mô hình thác nước và mô hình tiếp cận lặp và nó có nhiều điểm giống nhau với mô hình gia tăng.  +Mỗi giai đoạn trong mô hình được bắt đầu với yêu cầu/mục tiêu thiết kế và kết thúc với việc khách hàng kiểm tra tiến độ của từng giai đoạn. | +Lượng phân tích rủi ro cao.Do đó việc tránh rủi ro được tăng cường.  +Ước lượng được chi phí dễ dàng như việc hoàn thành 1 prototype trong một fragment nhỏ.  +Ứng dụng tốt đối với các dự án lớn và quan trọng.  +Kiểm soát tài liệu và phê duyệt chặt chẽ.  +Chức năng bổ sung hoặc thay đổi có thể được thêm vào những giai đoạn sau.  +Phần mềm sẽ được sản xuất sớm trong vòng đời của phần mềm.  +Ứng dụng được phát triển nhanh và các tính năng được thêm vào một cách có hệ thống.  +Luôn có thời gian cho khách hàng để phản hồi. | + Đối với rủi ro, ở giai đoạn phân tích cần một chuyên gia có chuyên môn cao để thực hiện việc phân tích.  + Không hữu ích với dự án có quy mô nhỏ.  + Thời gian và chi phí cho dự án có thể là vô hạn vì đặc tính xoắn ốc của mô hình.  +Tài liệu cho dự án có thể rất dài vì có các giai đoạn trung gian.  +Rủi ro có thể không đáp ứng được tiến độ hoặc ngân sách.  +Sự thành công của dự án phụ thuộc rất nhiều vào giai đoạn phân tích rủi ro. | +Khi dự án có quy mô lớn.  +Khi việc đánh giá  (phân tích) các chi phí và các rủi ro là quan trọng.  +Bất cứ lúc nào cũng có thể có yêu cầu realease thường  xuyên.  +Yêu cầu không rõ ràng và phức tạp.  Đối với các dự án có độ rủi ro từ trung bình đến cao.  +Những người sử dụng không chắc chắn về các nhu cầu  của họ.  +Các yêu cầu phần mềm phức tạp và lớn.  +Cần phát triển một dòng sản phẩm mới  +Khi có các thay đổi quan trọng. |
| Mô hình dựa thành phần | +Xây dựng hệ thống từ việc tích hợp các thành phần đang có hoặc các thành phần thương mại COTS. | +Mô hình dựa thành phần giảm 70% thời  gian và 84% chi phí.  +Phân tích thành phần có sẵn.  +Thiết kế hệ thống với kĩ thuật tái sử dụng. | +Sự phức tạp.  +Yếu tố con người. | +Các dự án nhỏ và ngắn hạn. |

**Bài 1.4:**

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| 1, Thông tin cá nhân:   * Họ tên * CMND/CCCD * Ngày cấp/Nơi cấp * Địa chỉ * Số điện thoại * Email | + Dữ liệu sẽ được đưa vào table chứa thông tin cá nhân của người làm đơn. | + Thông tin hiện ra sẽ bao gồm : thông tin người đăng kí và số lượng tài sản đăng kí |
| 2, Đăng kí sự kiện:   * Tên người đăng kí * Thời điểm đăng kí * Ngày đăng kí * Lí do * Khu vực tổ chức | + Dữ liệu sẽ được đưa vào table chứa thông tin cá nhân của người đăng kí. |  |
| 3, Trưng dụng tài sản   * Bàn,…… | + Số lượng tài sản được dùng sẽ tương tác trực tiếp với số lượng tài sản hiện có trong table chứa số tài sản chung. |  |