**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**──────── \* ───────**

**BÁO CÁO HỌC PHẦN**

**NHẬP MÔN CÔNG NGHỆ PHẦN MỀM**



**TÊN ĐỀ TÀI**

**BÀI TẬP LAB 2**

**NHÓM 9**

**Giảng viên hướng dẫn : Lương Mạnh Bá**

**Sinh viên thực hiện : Ngô Trung Hiếu – 20210145P**

**Đỗ Ngọc Hiếu – 20210144P**

**Nguyễn Quang Anh – 20210107P**

**Bùi Minh Chiến-** **20210117P**

**Phạm Tuấn Anh - 20210697P**

**Trần Tuấn Linh- 20210165P**

**Lớp : CNTT 1.2 - K66**

**Hà Nội 04-2023**

**Bài TH 02**

**NỘI DUNG 1: Phương pháp Agile &  Các hoạt động của Agile**

**Mục tiêu**

* Thực hiện các bài tập (câu hỏi) về nội dung Phương pháp Agile
* Phân tích các yêu cầu chức năng cho bài toán (case study)
* Bài tập về 4 hoạt động chính của Agile

**Đánh giá**

* Hoàn thành  các bài tập về nội dung Phương pháp Agile, nắm được tư tưởng của phương pháp Agile và các hoạt động chính. Phân biệt phương pháp Agile với các mô hình phát triển phần mềm khác
* Xác định các yêu cầu chức năng chính cho bài toán (case study) xây dựng được biểu đồ phân cấp chức năng và liệt kê một số yêu cầu phi chức năng
* Hoàn thành bài tập về 4 hoạt động chính của Agile

**Phần I:**

**Bài 1.1**

1. Nội dung nào sau đây được chuyển giao vào cuối Sprint?
2. Một tài liệu chứa các trường hợp kiểm thử cho sprint hiện tại
3. *Một thiết kế kiến trúc của giải pháp*
4. Một phiên bản gia tăng của phần mềm hoàn thành
5. Thiết kế giao diện người dùng

Đáp án: 1

1. Product Backlog chứa những thông tin cơ bản gì?
2. *Danh sách đầy đủ các yêu cầu hiện không có trong bản phát hành sản phẩm*
3. Danh sách nhật ký các hoạt động của nhóm phát triển sản phẩm
4. Danh sách các trường hợp kiểm thử
5. Danh sách các sơ đồ thiết kế của sản phẩm

Đáp án: 2

1. Chuỗi hoạt động nào sau đây là đúng cho Extreme Programming (XP)?
2. Write code, write test, refactor
3. Write code, refactor, write test
4. *Write test, write code, refactor*
5. Design, write code, write test

Đáp án: 2

**Bài 1.2**

Hãy so sánh mô hình thác nước (Waterfall model) và phương pháp Scrum:

|  |  |  |
| --- | --- | --- |
|  | **Mô hình thác nước**  **(Waterfall modell)** | **Phương pháp Scrum** |
| **Đặc điểm chính** | Các giai đoạn trong vòng đời được cấu trúc dạng “xuống dốc” từ trái quá phải. | Mỗi nhóm người thường xuyên gặp mặt trao đổi trong quá trình phát triển phần mềm. |
| **Đặc điểm các pha phát triển** | 1. Phân tích yêu cầu: - Nắm được các yêu cầu   2. Thiết kế hệ thống - Theo yêu cầu   3. Thực hiện - Tạo ra chương trình   4. Thử nghiệm hệ thống - Kiểm tra hoạt động hệ thống  5. Triển khai Hệ thống - Đảm bảo hệ thống hoạt động ổn định, ít gặp sự cố  6. Bảo trì hệ thống - Duy trì hoạt động của hệ thống và sửa lỗi nếu có. | 3 yếu tố cốt lõi của Scrum:   * Minh bạch * Thanh tra * Thích nghi   - Dự án chia thành các Sprint  - Mỗi sprint kéo dài từ 1-3 tuần  - Vào cuối mỗi Sprint, tất cả các bên liên quan họp nhau để đánh giá chi tiết tiến trình kế hoạch cho giai đoạn tiếp theo. |
| **Kích thước nhóm phát triển (team size)** | - Nhóm nhỏ và dự án nhỏ - Thời gian dự án ngắn | - Chia dự án cho nhiều nhóm nhỏ - Thời gian dự án có thể kéo dài |
| **Phong cách quản lý (management style)** | - Đơn giản, dễ hiểu   - Chính xác, rõ ràng   - Tuần tự, cứng nhắc | - Bao quát, minh bạch  - Tự giám sát  - Linh hoạt, thích nghi |
| **Quan điểm về sự thay đổi trong dự án** | - Khó có thể chấp nhận sự thay đổi yêu cầu  - Nếu có sự thay đổi thì việc quay lại cũng rất khó khăn | - Nếu có sự thay đổi, thay đổi chỉ diễn ra trong quy mô nhóm nhỏ  - Sự thay đổi sẽ được bàn bạc nhanh chóng do nhóm thường xuyên trao đổi với nhau. |
| **Vấn đề xây dựng tài liệu** | - Kết quả được ghi chép tốt, tài liệu qua từng giai đoạn. - Tài liệu có thể xây dựng tường minh và dễ sử dụng | - Các tài liệu cần thiết trong Scrum như: Product Backlog, Sprint Backlog và Product Increment là những công cụ hỗ trợ công việc rất hiệu quả. |
| **Đối phó với các nguy cơ / rủi ro (risk)** | - Khó lường trước nguy cơ rủi ro nên phải khắc phục rủi ro bằng cách dừng toàn bộ   - Khó khăn trong việc di chuyển trở lại giai đoạn cần thiết | - Do đã chia dự án thành các Sprint nên quy mô gặp phải rủi ro nhỏ hơn nên việc khắc phục rủi ro dễ dàng hơn.   - Có thể chia ra một số module có nguy cơ lỗi cao để kiểm thử dễ dàng hơn. |
| **Tình huống áp dụng phù hợp** | - Ứng dụng nhỏ   - Môi trường, công nghệ và công cụ ổn định   - Nguồn lực được đào tạo và luôn sẵn sàng | - Phát triển những dự án mơ hồ và phức tạp  - Nhưng dự án tồn tại những module có nguy cơ gặp lỗi cao |

**Phần II: Bài tập về 4 hoạt động chính của Agile**

Nêu 4 hoạt động chính và phân tích (slide)

**1. Daily Meeting:**

- Tất cả các thành viên báo cáo công việc hàng ngày

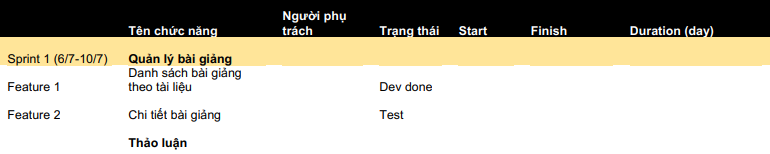
**2. Sprint Planning:**

- Lên kế hoạch công việc làm tỏng 1 sprint

**3. Sprint Review:**

- Xem xét lại các công việc đã làm trong sprint, phần nào có thể release/deliverable

Dưới đây là 1 template ví dụ cho 3 hoạt động trên:



Ảnh có chứa bàn

Mô tả được tạo tự động

**4. Sprint Retrospective:**

- Tổng kết điểm mạnh/ yếu, điểm cần chỉnh sửa trong công việc (mọi người đều nói)

Ví dụ:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Điểm mạnh**  **(Đã làm tốt)** | **Điểm yếu                     (Chưa làm tốt)** | **Cần chỉnh sửa**  **(Học hỏi thêm)** |
| Nguyen Quang Anh | \_ Teamwork tốt  \_ Trao đổi giữa team A và team B tốt, nhiệt tình | \_ Chưa xác định thời gian hoàn thành  cho từng task cụ thể | \_ Xác định thười gian hoàn thành cụ thể cho từng task, tạo grant chart để quản lý tiến độ các task và plan |
| Phạm Tuấn Anh | \_ Học hỏi nhanh  \_ Dùng các tool tốt  \_ Code nhanh gọn dễ hiểu | \_ Code chưa viết test | \_ Học thêm Docker |
| Ngô Trung Hiếu | \_ Làm việc với nhiều công nghệ: Lambda, IAM, RDS, ECS… | \_ Cần hoàn thành đúng thời gian được giao | \_ Làm việc nhiều hơn với team B Backend side |

**NỘI DUNG 2: Quản lý dự án phần mềm & Lập trình với giao diện đồ hoạ người dùng (GUI)**

**Mục tiêu**

* Thực hiện các bài tập (câu hỏi) về nội dung Quản lý dự án phần mềm
* Phân rã các chức năng đã xác định thành các nhiệm vụ (task) để xây dựng một kế hoạch dự án phần mềm đơn giản
* Lập trình với giao diện đồ hoạ người dùng:
  + Làm quen với mô hình MVC các thành phần giao diện người dùng
  + Thiết kế giao diện GUI trên công cụ Netbeans
  + Viết mã xử lý sự kiện

**Đánh giá**

* Hoàn thành  các bài tập về nội dung Quản lý dự án phần mềm, cơ bản nắm được nguyên lý, quy trình và các kỹ năng / kỹ thuật trong quản lý dự án
* Các nhóm sinh viên xây dựng được biểu đồ Cấu trúc phân chia công việc (Work Breakdown Structure - WBS)
* Hoàn thành bài thực hành lập trình với giao diện đồ hoạ người dùng

**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

Đáp án: 1,2,3,5

1. Một dự án phát triển phần mềm là thành công khi?
2. Sản phẩm đáp ứng yêu cầu chất lượng
3. Không vượt quá kinh phí dự kiến
4. Hoàn thành trong thời gian dự kiến
5. Tất cả các phương án trên

Đáp án: 4

1. … 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.
2. Rủi ro dự án
3. Lập kế hoạch dự án
4. Quản lý nhân sự
5. Yêu cầu người dùng

Đáp án: 1

**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?

Bg

-  Đối với việc một người làm công việc tham gia vào một dự án phát triển phần mềm mà không biết đầy đủ về chức năng của dự án, thì đó là điều hợp lý.

- Thực tế là trong các dự án phát triển phần mềm lớn, phức tạp, các thành viên trong nhóm thường được giao phụ trách các nhiệm vụ riêng lẻ, thường chỉ là một phần của dự án lớn hơn.

- Các dự án phần mềm hiện đại thường sử dụng các nguyên tắc của phân phối, vì vậy một dự án sẽ được phân chia thành nhiều mô-đun nhỏ hơn, và các phần tử của nhóm chỉ được xác định cho từng mô-đun để phát triển. Mỗi người trong nhóm sẽ tập trung vào việc phát triển mô-đun của mình, và chỉ cần biết đủ về các thành phần liên quan đến nó mà không cần hiểu quy trình và chức năng của toàn bộ dự án.

- Tuy nhiên, để chắc chắn rằng các thành viên trong nhóm hiểu rõ mục tiêu của dự án và cách thức đóng góp đóng góp tích cực vào dự án, họ cần đưa ra giả thuyết được định nghĩa rõ ràng của mục tiêu dự án, những người yêu thích cầu và kế hoạch của dự án. Họ cũng nên được đào tạo để hiểu cách làm việc đúng tiến độ, tuân thủ các tiêu chuẩn được sử dụng cho dự án và cách thức thực hiện công việc thành công trong môi trường của dự án.

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

              Bài giải

-  Có thể phát triển phần mềm mà không áp dụng các mô hình vòng đời phần mềm.

- Tuy nhiên, việc sử dụng các mô hình vòng đời của phần mềm được khuyến khích và rất quan trọng trong việc quản lý dự án phát triển phần mềm.

- Các mô hình vòng đời phần mềm như mô hình Thác nước, Agile, Xoắn ốc và Lặp lại đều cung cấp khuôn khổ cho quy trình phát triển phần mềm. Các mô hình này định nghĩa các giai đoạn/đợt, tài liệu và yêu cầu cần phải có trước khi chuyển sang giai đoạn/đợt tiếp theo.

-  Việc sử dụng các mô hình vòng đời Phần mềm giúp dự án thực hiện theo kế hoạch, đưa ra các mức tiêu chuẩn, giảm thiểu rủi ro, liên tục kiểm tra chất lượng sản phẩm và có thể đáp ứng yêu cầu của khách hàng một cách toàn diện hơn.

-  Nếu không áp dụng các mô hình vòng đời phần mềm, thì có thể dẫn đến việc phát triển phần mềm không đáp ứng được yêu cầu của khách hàng hoặc không đạt được chất lượng như mong đợi. Việc không đạt được tiến độ và chi phí của dự án là điều tất yếu khi thiếu quy định quản lý dự án rõ ràng. Ngoài ra, việc không sử dụng các mô hình vòng đời phần mềm cũng dẫn đến khó khăn trong việc quản lý các thiết kế, mã nguồn và tài liệu cho dự án.

- Tóm lại, việc sử dụng các mô hình vòng đời phần mềm không phải là điều kiện bắt buộc, nhưng nó sẽ giúp quá trình phát triển phần mềm được quản lý, tiếp cận một cách có hệ thống và làm cho sản phẩm phần mềm phát triển phát triển đạt được hiệu quả, chất lượng và độ chính xác cao hơn.

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

               Bài giải

-  Trong phương pháp Agile, việc làm luôn có đại diện của khách hàng trong nhóm phát triển có nhiều ưu điểm, bao gồm:

-  Giúp đảm bảo sản phẩm phát triển phát triển đáp ứng nhu cầu của khách hàng: Với người đại diện của khách hàng luôn có mặt trong nhóm phát triển, sản phẩm sẽ luôn được xây dựng với sự hỗ trợ của người đại diện khách hàng, giúp đảm bảo khẳng định rằng sản phẩm đang phát triển đang đáp ứng nhu cầu của khách hàng.

-  Giúp đảm bảo sự tương tác giữa khách hàng và nhóm phát triển: Người đại diện của khách hàng luôn ở gần và có liên lạc thường xuyên với nhóm phát triển, điều này giúp giảm thiểu rủi ro về công việc thông tin được truyền đạt sai hoặc mất mát thông tin giữa khách hàng và nhóm phát triển.

-  Giúp sản phẩm được phát triển nhanh hơn và chính xác hơn: Với việc có người đại diện của khách hàng đồng lòng với nhóm phát triển về các sản phẩm được yêu cầu, các quyết định có thể được đưa ra nhanh chóng, giảm thiểu sự chậm chạp thấy trong sự việc phát triển và chắc chắn rằng sản phẩm được phát triển chính xác hướng đến đúng mục tiêu.

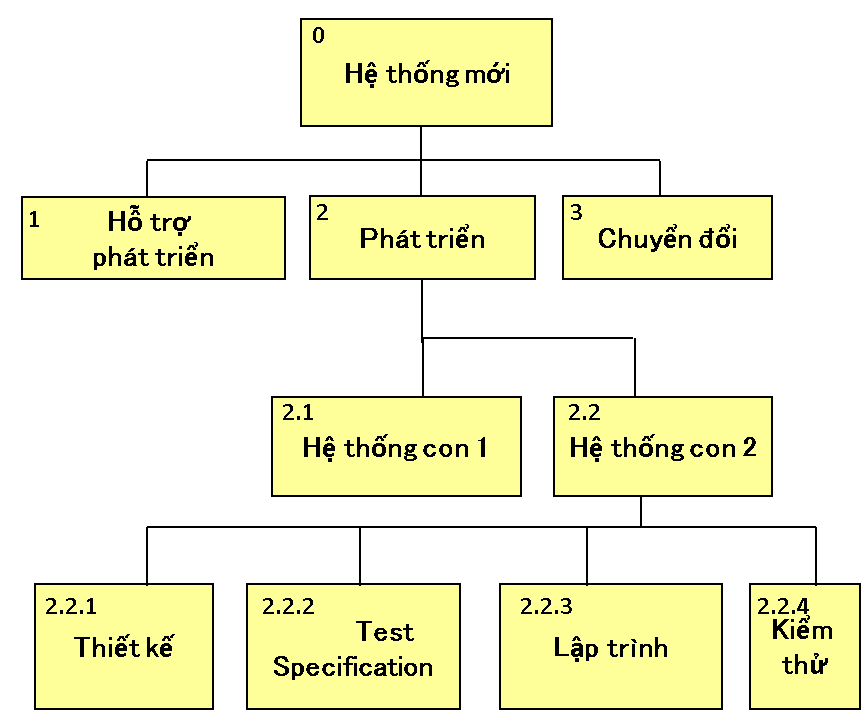
-  Giúp tăng tính hoạt động và sự thích nghi của quá trình phát triển sản phẩm: Một trong những điểm ưu tiên nhất của phương pháp Agile là sự dẻo dai và sự thích nghi cao. Nếu có người đại diện của khách hàng, nhóm phát triển có thể hài lòng và điều chỉnh yêu cầu sản phẩm bất cứ lúc nào trong quá trình phát triển. Điều này giúp sản phẩm được yêu thích với thị trường và nhu cầu thay đổi của khách hàng hiệu quả hơn.

 ->  Việc luôn có đại diện của khách hàng trong nhóm phát triển theo phương pháp Agile Hỗ trợ tốt cho việc phát triển sản phẩm, giúp đảm bảo sự xây dựng sản phẩm chính xác và đáp ứng nhu cầu của khách hàng, tăng trưởng tính linh hoạt và sở thích của quá trình phát triển sản phẩm.

**Phần II: Xây dựng biểu đồ WBS**

* WBS là một phân rã phân cấp (hierarchical decomposition) của toàn bộ phạm vi công việc sẽ được nhóm dự án thực hiện để hoàn thành các mục tiêu của dự án và tạo ra các sản phẩm bàn giao được yêu cầu.
* Các thành phần phân rã ở mức cuối cùng – mức lá nên thoả mãn các tiêu chí:
  + Tình trạng / tính hoàn tất của công việc có thể đo được hoặc có sản phẩm cụ thể
  + Thời gian, tài nguyên / chi phí có thể ước lượng được
  + Thời gian hoàn thành công việc trong giới hạn
  + Công việc được phân công độc lập (nghĩa là công việc không bị ngừng giữa chừng để chờ kết quả của công việc khác)

- Ví dụ:



1. **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.**

Trên cơ sở biểu đồ WBS phía trên hãy xây dựng một bản kế hoạch đơn giản như sau:

|  |  |  |
| --- | --- | --- |
| **Công việc** | **Thời gian (số giờ làm việc)** | **Số người** |
| **1. Công việc 1**  **1.1 Công việc 1.1**  **1.2 Công việc 1.2**  **…** |  |  |
| **2. Công việc 2**  **2.1 Công việc 2.1**  **2.2 Công việc 2.2**  **…** |  |  |
| **…** |  |  |

Để có thể làm ra một phần mềm quản lý thông tin phản ánh, kiến nghị của Ban quản lý tổ dân phố 7 phường La Khê, chúng ta có thể sử dụng quy trình phát triển  phần mềm theo chu trình Agile với các bước sau :

     Bước 1 : Thu thập yêu cầu và phân tích yêu cầu

Trong bước này, chúng ta sẽ tiến hành trao đổi với Ban quản lý tổ dân phố 7 để có thể thu thập và hiểu rõ hơn về những yêu cầu của họ. Sau đó, chúng ta sẽ cùng phân tích yêu cầu để tạo ra một tài liệu chi tiết. Tài liệu này sẽ gồm các mục sau :

* Các mô tả tổng quan về phần mềm quản lý thông tin, phản ánh kiến nghị của Ban quản lý tổ dân phố 7.
* Các tính năng cần có của phần mềm và các yêu cầu kỹ thuật cho từng tính năng.
* Các ràng buộc và giới hạn về dữ liệu, bảo mật, hiệu suất và tương thích hệ thống.

     Bước 2 :  Thiết kế hệ thống

 Dựa trên tài liệu yêu cầu chi tiết, chúng ta sẽ tiến hành thiết kế hệ thống. Trong bước này,

thiết kế các đối tượng, các module, các chức năng, các giao diện và các quy trình cho phần mềm.

     Bước 3 :  Phát triển và lập trình

  Dựa trên bộ thiết kế hệ thống ở phần trên chúng ta đã có đầy đủ các thông tin

và các yêu cầu kỹ thuật về các tính năng, chúng ta sẽ bắt đầu phát triển và viết các đoạn mã cho phần mềm

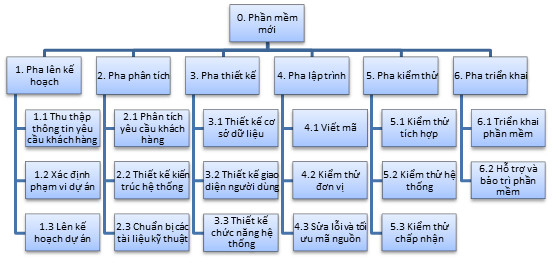
Bước 4 : Kiểm thử phần mềm :

Trong quá trình phát triển phần mềm, chúng ta sẽ tiến hành kiểm thử liên tục để đảm bảo

phần mềm hoạt động tốt và đáp ứng được các yêu cầu của Ban quản lý tổ dân phố 7.

      Bước 5 : Triển khai và vận hành

Sau khi kiểm thử và phát triển, chúng ta sẽ triển khai phần mềm cho ban quản lý tổ dân phố 7. Trong quá trình triển khai, chúng ta sẽ đảm bảo rằng phần mềm hoạt động tốt trên môi trường thực tế. Sau khi triển khai xong, chúng ta vẫn sẽ vận hành và bảo trì phần mềm  thường xuyên để đảm bảo sự ổn định và đáp ứng được các nhu cầu của Ban quản lý tổ dân phố khu 7



|  |  |  |
| --- | --- | --- |
| **Công việc** | **Thời gian**  **(Số giờ làm việc)** | **Số người** |
| 1 . Pha lên kế hoạch | 40 | 2 |
| 1.1. Thu thập thông tin | 16 | 1 |
| 1.2. Xác định phạm vị | 8 | 1 |
| 1.3. Lên kế hoạch dự án | 16 | 1 |
| 2. Pha phân tích | 64 | 3 |
| 2.1. Phân tích yêu cầu | 32 | 2 |
| 2.2. Thiết kế kiến trúc | 16 | 1 |
| 2.3 Chuẩn bị tài liệu | 16 | 1 |
| 3. Pha thiết kế | 80 | 4 |
| 3.1. Thiết kế cơ sở dữ liệu | 24 | 1 |
| 3.2. Thiết kế giao diện | 32 | 2 |
| 3.3. Thiết kế chức năng | 24 | 1 |
| 4. Pha lập trình và phát  triển | 160 | 4 |
| 4.1  Viết mã | 80 | 2 |
| 4.2. Kiểm thử đơn vị | 40 | 1 |
| 4.3. Sửa lỗi và tối ưu mã nguồn | 40 | 1 |
| 5. Kiểm thử | 128 | 3 |
| 5.1. Kiểm thử tích hợp | 32 | 2 |
| 5.2. Kiểm thử hệ thống | 64 | 1 |
| 5.4. Kiểm thử chấp nhận | 32 | 1 |
| 6. Pha triển khai | 56 | 2 |
| 6.1 Triển khai phần mềm | 32 | 1 |
| 6.2. Hỗ trợ và bảo trì phần mềm | 24 | 1 |

1. **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.**

Ví dụ:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Công việc / Hoạt động** | **Xác định rủi ro** | | | **Quản lý rủi ro** | |
| **Mối nguy** | **Rủi ro** | **Mức độ** | **Chiến lược** | **Biện pháp** |
| Mua hàng | Hàng bị hư hại khi vận chuyển | Không có nguyên liệu sản xuất | Trung bình | Giảm thiểu | Xây dựng tồn kho tối thiểu |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

(Mức độ có thể đánh giá theo: thấp / trung bình / cao)

(Chiến lược có thể lựa chọn: Giảm thiểu rủi ro / Chuyển rủi ro / Loại bỏ rủi ro)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 | Thiếu thông tin yêu cầu từ khách hàng | Không xác định được yêu cầu | Trung bình | Chuyển rủi ro | Họp trực tiếp với khách hàng để làm rõ yêu cầu, xác định yêu cầu rõ ràng. Đa dạng hóa nguồn lực, xây dựng đội ngũ chuyên gia |
| Thiết kế kiến trúc hệ thống | Không thể thiết kế được | Không đủ kinh nghiệm về công nghệ mới | Cao | Loại bỏ rủi ro | Thuê chuyên gia để tư vấn, đào tạo nhân viên. Tuyển dụng thêm nhân viên có kinh nghiệm, đào tạo nhân viên hiện có |
| Phát triển phần mềm | Sai sót trong việc lập trình | Thiếu nguồn lực cao, chất lượng | Cao | Loại bỏ rủi ro | Tối ưu hóa phương pháp làm việc, xây dựng kế hoạch rõ ràng. Thuê thêm nhân viên, sử dụng công cụ hỗ trợ phát triển |
| Kiểm thử hệ thống | Không đạt được kết quả mong đợi | Thiếu tài nguyên để kiểm thử | Trung bình | Giảm thiểu rủi ro | Áp dụng phương pháp kiểm thử tự động, tối ưu hóa kế hoạch kiểm thử để đảm bảo chất lượng sản phẩm. Định kỳ kiểm tra tình trạng tiến độ kiểm thử |
| Triển khai phần mềm | Không thể triển khai được | Thiếu kế hoạch triển khai | Trung bình | Giảm thiểu rủi ro | Xây dựng kế hoạch triển khai chi tiết, loại bỏ rủi ro có thể gây ra |
| Hỗ trợ và bảo trì phần mềm | Khách hàng không hài lòng với dịch vụ hỗ trợ | Khách hàng không sử dụng dịch vụ hoặc hủy bỏ dịch vụ | Trung bình | Giảm thiểu rủi ro | Đào tạo nhân viên hỗ trợ để có thể giải quyết các vấn đề của khách hàng nhanh chóng và hiệu quả hơn. Giảm thiểu rủi ro của phần mềm và đào tạo hỗ trợ khách hàng |

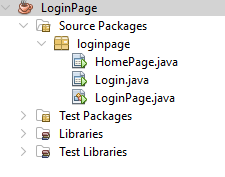
**\* Phần III SV hệ vừa học vừa làm không phải làm**

**Phần III: Thực hành lập trình giao diện đồ hoạ người dùng GUI**

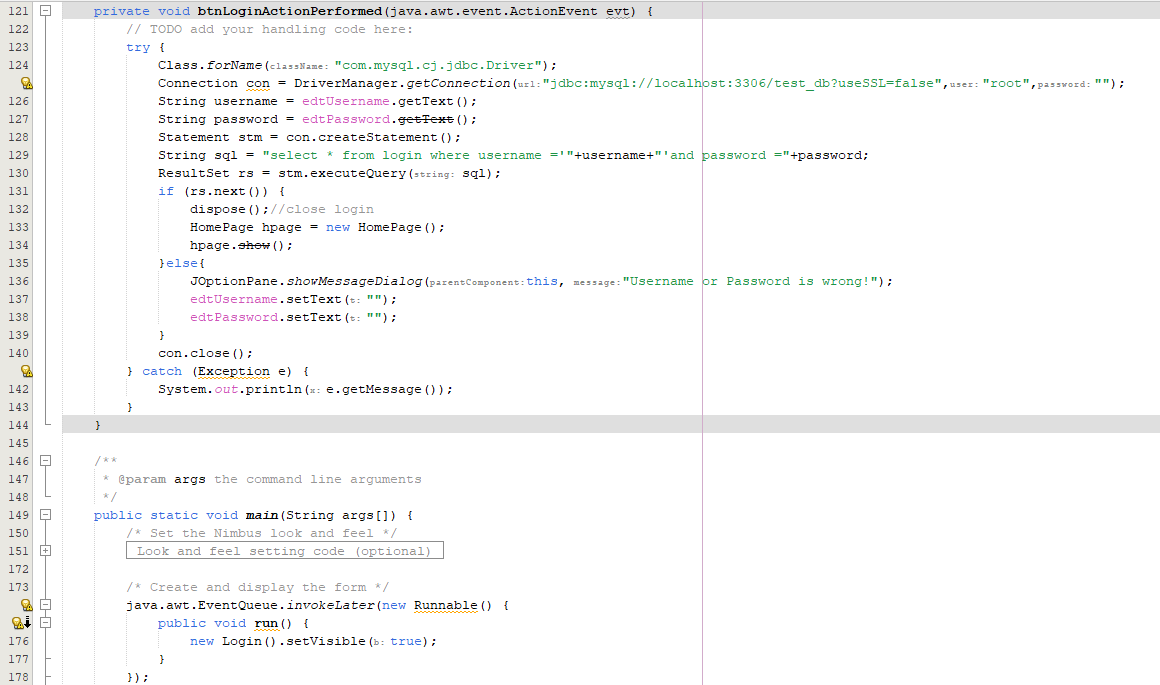
**Nội dung bài tập tự làm**

* Thực hành xây dựng chương trình java với giao diện đồ hoạ: Liệt kê danh sách người dùng sau khi đăng nhập thành công:

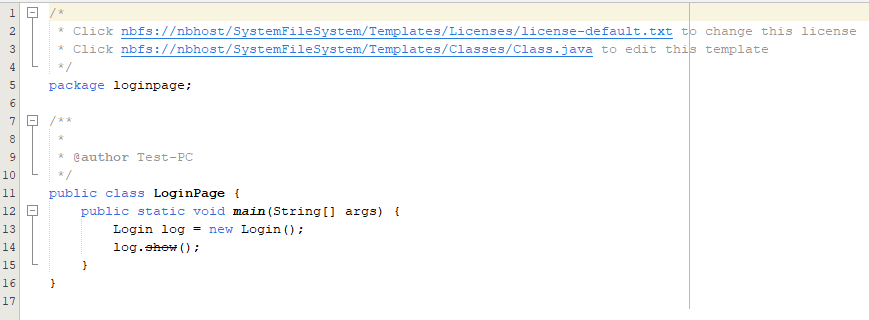
**Mã nguồn**

****

****

****

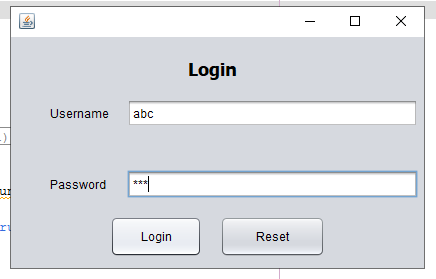
****

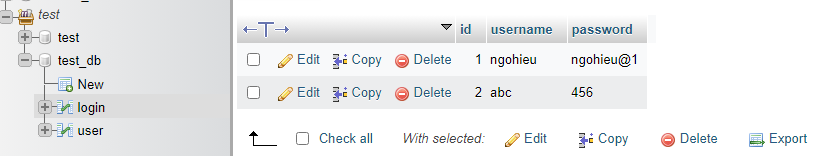
****

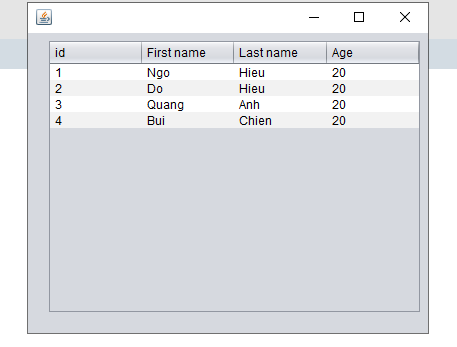
****

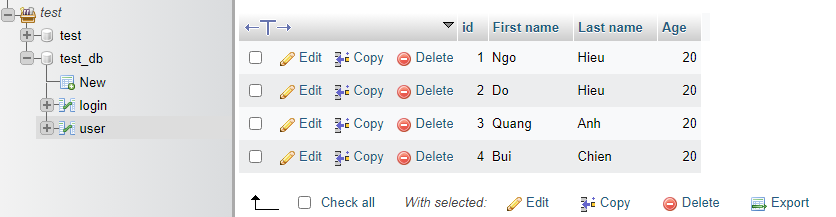
****

**\* Chạy chương trình**

****

****

****

****

**HẾT**