Bài tập tuần 03

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
6. Product Backlog chứa những thông tin cơ bản gì?
7. Danh sách đầy đủ các yêu cầu hiện không có trong bản phát hành sản phẩm
8. Danh sách nhật ký các hoạt động của nhóm phát triển sản phẩm
9. Danh sách các trường hợp kiểm thử
10. Danh sách các sơ đồ thiết kế của sản phẩm
11. Chuỗi hoạt động nào sau đây là đúng cho Extreme Programming (XP)?
12. Write code, write test, refactor
13. Write code, refactor, write test
14. Write test, write code, refactor
15. Design, write code, write test

Bài 1.2

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

|  |  |  |
| --- | --- | --- |
|  | Mô hình thác nước (Waterfall modell) | Phương pháp Scrum |
| Đặc điểm chính | Phương pháp mô hình thác mà còn được gọi là mô hình vòng tuần hoàn dạng vòng lặp. Mô hình thác nước theo thứ tự tuần tự và do đó nhóm phát triển dự án chỉ chuyển sang giai đoạn phát triển hoặc thử nghiệm tiếp theo nếu bước trước đó hoàn thành thành công. | Scrum là một khung quản lý được các đội ngũ sử dụng để tự tổ chức và hoạt động vì một mục tiêu chung. Khung này mô tả một loạt các cuộc họp, công cụ và vai trò để bàn giao dự án hiệu quả. Cũng giống như một đội thể thao tập dượt cho trận đấu quan trọng, thực hành Scrum tạo điều kiện cho đội ngũ tự quản lý, học hỏi từ kinh nghiệm và thích nghi với sự thay đổi. Các đội ngũ phát triển phần mềm sử dụng Scrum để giải quyết những vấn đề phức tạp một cách tiết kiệm và bền vững. |
| Đặc điểm các pha phát triển | Chuỗi các hoạt động qua các giai đoạn của vòng đời “xuống dốc”: phân tích, yêu cầu, đặc tả, thiết kế, cài đặt, kiểm thử, bảo trì | * Quá trình phát triển được chia thành một loạt các lần lặp được gọi là sprint. * Trước mỗi sprint, các thành viên trong nhóm xác định các công việc còn tồn. * Khi kết thúc sprint, nhóm sẽ xem xét để nêu rõ các kinh nghiệm và kiểm tra tiến độ |
| Kích thước nhóm phát triển (team size) | Thích hợp cho các dự án lớn, đội ngũ nhân viên đông, dày dặn kinh nghiệm | Jeff Bezos – nhà sáng lập của Amazon – đã đề xuất quy tắc hai chiếc bánh pizza khi quyết định quy mô của đội ngũ: Một đội ngũ chỉ nên vừa đủ để chia đều hai chiếc bánh pizza. Kích thuóc chỉ tầm 16 người – nhóm nhỏ. |
| Phong cách quản lý (management style) | Phương pháp quản lí dự án dựa trên qui trình thiết kế tuần tự và liên tiếp. | Scrum sử dụng Sprint nhỏ để chia hệ thống thành các thành phần nhỏ hơn một cách hiệu quả và phân chia cho các nhóm. |
| Quan điểm về sự thay đổi trong dự án | Khó thích ứng với thay đổi bao gồm yêu cầu, kế hoạch, phạm vi dự án. | Cho phép cải tiến liên tục, do đó cho phép áp dụng kỹ thuật từ dưới lên một cách nhanh chóng |
| Vấn đề xây dựng tài liệu | Mô hình thác nước nên sử dụng khi mà đội dự án đã có kinh nghiệm làm việc, trình độ chuyên môn và kỹ thuật cao bởi mô hình này yêu cầu  sự chính xác ngay từ khi bắt đầu dự án. | Công ty phải có kiến thức về phương pháp làm việc trước khi sử dụng Scrum. |
| Đối phó với các nguy cơ / rủi ro (risk) | * Phụ thuộc vào các yêu cầu được xác định sớm từ đầu * Phụ thuộc vào việc tách các yêu cầu khỏi thiết kế * Không khả thi trong một số trường hợp đòi hỏi có nhiều thay đổi * Nhấn mạnh vào sản phẩm hơn là quy trình | * Scrum chỉ hoạt động khi người quản lý “tin tưởng các nhóm phát triển”. Scrum phụ thuộc vào khả năng kiểm soát, vì vậy, nếu đội ngũ phát triển còn trẻ và chưa trưởng thành thì Scrum trở nên rất rủi ro. * Scrum được thiết kế lý tưởng cho công ty có “các phương pháp nhanh hiện có”. Do đó, một công ty phải có kiến thức về phương pháp làm việc trước khi sử dụng Scrum. |
| Tình huống áp dụng phù hợp | Mô hình thác nước phù hợp với các dự án quy mô lớn, yêu cầu duy trì các giai đoạn và thời hạn theo đúng kế hoạch đã lập ra. Hoặc các dự án đã được thực hiện nhiều lần mà ít xảy ra phát sinh trong quá trình thực thi. Đặc biệt phù hợp trong dự án sản xuất và xây dựng các sản phẩm vật lý và theo các đơn đặt hàng, có thể dễ dàng tham khảo áp dụng các quy trình quản lý từ các dự án trước đó vào công việc hiện tại với rất ít hoặc không cần điều chỉnh. | Quy trình Scrum phổ biến trong ngành công nghệ và hiện đang được lựa chọn bởi tất cả các lĩnh vực khác. Tuy nhiên, không phải lúc nào các công ty cũng dễ dàng áp dụng. Là một phương pháp luận linh hoạt, Scrum thường bị các công ty có hệ thống phân cấp dọc nghiêm ngặt chống lại, nhưng lại được nhiều công ty khởi nghiệp thực hiện nhờ sự linh hoạt. |

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

1. Dailly meeting

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

1. Sprint Planning

* Lên kế hoạch công việc làm trong 1 sprint

1. Sprint Reiview

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

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Tên chức năng** | **Người phụ trách** | **Trạng thái** | **Start** | **Finish** | **Duration (day)** |
| Sprint 1 (6/7-10/7) | **Quản lý bài giảng** |  |  |  |  |  |
| Feature 1 | Danh sách bài giảng theo tài liệu |  | Dev done |  |  |  |
| Feature 2 | Chi tiết bài giảng |  | Test |  |  |  |
|  | **Thảo luận** |  |  |  |  |  |
| Feature 3 | Danh sách thảo luận |  |  |  |  |  |
| Feature 4 | Tạo câu hỏi mới, reply |  |  |  |  |  |
| Feature 5 | Tối ưu UI |  |  |  |  |  |
| Feature 6 | Fix bugs tồn đọng |  |  |  |  |  |
| Sprint 3 (20/7-24/7) | **Tối ưu** |  |  |  |  |  |
| Feature 1 | Tối ưu học phần |  | Doing |  |  |  |
| Feature 2 | Tối ưu danh sách lớp học |  | Doing |  |  |  |
| Feature 5 | Tối ưu UI |  | Doing |  |  |  |
| Feature 6 | Xử lý các task phát sinh |  | Doing |  |  |  |
|  |  |  |  |  |  |  |

1. Restrospective

* Tổng kết điểm mạnh/yếu, điểm cần chỉnh sửa trong công việc (mọi người đều nói)
* **Ví dụ1:**

1. Nguyen Van A (Leader)

### Positive

* Complete converting from python 2 to python 3 for Coupon.
* Understanding more detail about writing test case.

### Negative

- Skill for writing test case not good. Take long time for writing test case.

### Review of Last Month's Try

* More understanding about clean code.

### Next Month's Try

* Improve programing skill of kiwi members. Read book: <https://www.amazon.co.jp/Art-Readable-Code-Practical-Techniques/dp/0596802293>
* Train new bridge SE for understanding system => cần output và quá trình. Phần này cần xong vào tuần sau.

2. Nguyen Van B (dev)

### Positive

* Complete action of message list

### Negative

* Too long of review time

### Last month's try

* Reference current design to apply my code to reduce review time (Continue)

### Revew of Last Month's Try

* Review time is still long

### Next month's try

* Confirm design before code
* Add comment to MR to explain the intention
* **Ví dụ 2:**

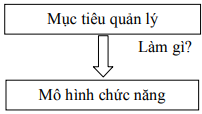
|  |  |  |  |
| --- | --- | --- | --- |
|  | Good (đã làm được gì, học hỏi được gì, ...) | Bad (có gì không vừa ý, làm chưa tốt, ...) | Try (có thể cải thiện thế nào, mong muốn, đề xuất, ...) |
| Nguyen Van A | Teamwork tốt | Chưa xác định thời gian hoàn thành cho từng task cụ thể | Xác định thời gian hoàn thành cụ thể cho từng task, tạo grant chart để quản lý tiến độ các task và plan |
|  | Trao đổi giữa team A và team B tốt |  |  |
|  | Team B review và support nhiệt tình |  |  |
|  | Kế hoạch làm việc rõ ràng, cụ thể |  |  |
| Nguyen Van B | - học thêm docker |  |  |
|  | - cơ hội dùng tool redash | Không hài lòng với tool Redash |  |
| Nguyen Van C | Làm việc với nhiều công nghê: | Code chưa viết test | Tiếp tục tìm hiểu về Amazon Web Service |
|  | - AWS Fargate, SQS, Lambda, IAM, RDS, ECS |  | Hiều sâu về devops process |
|  | - Google cloud storage, credential |  | Làm việc nhiều hơn với Tem B Backend side |
|  | - Terraform , apex, docker |  |  |
|  | Làm việc devops : CI, CD , Automation |  |  |

Phần III: Phân tích chức năng cho bài toán (case study)

- Tiếp theo kết quả của bài tập tuần 02, các nhóm đã xác định được thông tin cơ bản: đầu vào (input), các xử lý chính (process) và kết quả (output).

- Biểu đồ phân cấp chức năng (BFD - Bussiness Function Diagram): phân rã có thứ bậc đơn giản các chức năng của hệ thống thành các chức năng nhỏ hơn, cuối cùng thu được một cây chức năng.

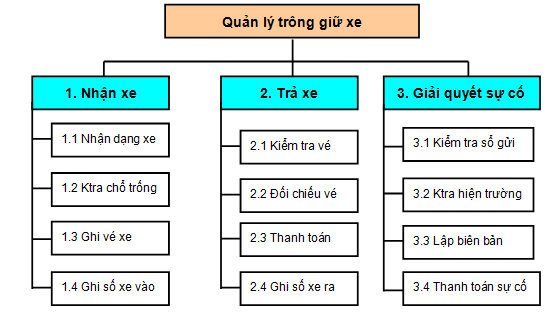
- Biểu đồ này thể hiện các đầu việc mà hệ thống cần thực hiện để đạt được mục tiêu quản lý (quản lý cái gì?), xây dựng dựa trên cơ sở của bước khảo sát và phân tích cơ bản trước đó.



- Các ký hiệu chính:

* Ký hiệu chức năng là một hình chữ nhật bên trong là tên chức năng, Tên chức năng là động từ - bổ ngữ và động từ nên ở dạng thức mệnh lệnh. Các chức năng cần đánh số thể hiện sự phân cấp.
* Liên kết các chức năng là đường thẳng

- Ví dụ:



Mô tả các chức năng trong biểu đồ:

|  |  |  |
| --- | --- | --- |
| Tên chức năng | Mô tả | Đánh giá khả năng thực hiện (nhân lúc, thời gian, công nghệ, môi trường) |
|  |  |  |
|  |  |  |

**Yêu cầu:** **Hãy xây dựng biểu đồ phân cấp chức năng cho cho nhóm yêu cầu nghiệp vụ của bài toán (case study) mà nhóm đã lựa chọn. Mô tả các chức năng đó.**

Nội dung bài tập tự làm

**HẾT**