**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**PHẠM ÁNH DƯƠNG**

**KHÓA LUẬN TỐT NGHIỆP**

**NGHIÊN CỨU MÔ HÌNH AGILE VÀ PHÁT TRIỂN CÔNG CỤ HỖ TRỢ**

**KỸ SƯ NGÀNH CÔNG NGHỆ PHẦN MỀM**

**TP. HỒ CHÍ MINH, 2014**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**

**PHẠM ÁNH DƯƠNG – 09520047**

**KHÓA LUẬN TỐT NGHIỆP**

**NGHIÊN CUU1 MÔ HÌNH AGILE VÀ PHÁT TRIỂN CÔNG CỤ HỖ TRỢ**

**KỸ SƯNGÀNH CÔNG NGHỆ PHẦN MỀM**

**GIẢNG VIÊN HƯỚNG DẪN**

**NGUYỄN THỊ THANH TRÚC**

**TP. HỒ CHÍ MINH, 2014**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số …………………… ngày ………………….. của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

* 1. …………………………………………. – Chủ tịch.
  2. …………………………………………. – Thư ký.
  3. …………………………………………. – Ủy viên.
  4. …………………………………………. – Ủy viên.

MỤC LỤC

[Chương 1. TÊN CHƯƠNG 1 3](#_Toc367742496)

[1.1. Chủ đề cấp độ 2 3](#_Toc367742497)

[1.1.1. Chủ đề cấp độ 3 3](#_Toc367742498)

[1.1.2. Chủ đề cấp độ 3 3](#_Toc367742499)

[1.1.2.1. Chủ đề cấp độ 4 3](#_Toc367742500)

[Chương 2. TÊN CHƯƠNG 2 4](#_Toc367742501)

[2.1. Chủ đề cấp độ 2 4](#_Toc367742502)

[2.1.1. Chủ đề cấp độ 3 4](#_Toc367742503)

[2.1.1.1. Chủ đề cấp độ 4 4](#_Toc367742504)

[2.2. Chủ đề cấp độ 2 4](#_Toc367742505)

[2.2.1. Chủ đề cấp độ 3 4](#_Toc367742506)

[Chương 3. TÊN CHƯƠNG 3 5](#_Toc367742507)

[3.1. Chủ đề cấp độ 2 5](#_Toc367742508)

[3.1.1. Chủ đề cấp độ 3 5](#_Toc367742509)

[3.1.1.1. Chủ đề cấp độ 4 5](#_Toc367742510)

[3.2. Chủ đề cấp độ 2 5](#_Toc367742511)

DANH MỤC HÌNH VẼ

[Hình 1.1: Tên hình 1 3](#_Toc367742554)

DANH MỤC BẢNG

[Bảng 1.1: Tên bảng 1 3](#_Toc367742567)

[Bảng 2.1: Tên bảng 1 4](#_Toc367742568)

DANH MỤC TỪ VIẾT TẮT

TÓM TẮT KHÓA LUẬN

Agile nổi lên như một triết lí phát triển phần mềm mới mẻ với cách tiếp cận khác khắc phục những điểm yếu của các phương pháp phát triển phần mềm truyền thống. Agile là một tập các phương thức phát triển phần mềm theo hướng lặp và tăng dần, trong đó Scrum nổi bật lên là phương pháp được sử dụng nhiều nhất.

Khóa luận tập trung tìm hiểu khái niệm, đặc trưng, cách hiện thực của Agile cùng một phương pháp nổi tiếng của Agile là Scrum. Trong quá trình tìm hiểu, nhóm nhận thấy Scrum là phương pháp dễ tiếp cận phù hợp với quy mô đa dạng các dự án phần mềm nên nhóm đã tập trung vào xây dựng công cụ hỗ trợ quản lí dự án công nghệ thông tin theo mô hình Scrum.

Ứng dụng bám sát vào quy trình phát triển phần mềm theo phương pháp Scrum. Với giao diện trực quan, tiện lợi và cập nhật thời gian thực, ứng dụng nhằm giúp người dùng cảm thấy tiện lợi nhất khi sử dụng, và dễ dàng hơn khi xây dựng sản phẩm theo phương pháp Scrum.

MỞ ĐẦU

Trong ngành công nghệ phần mềm ờ thời kì ban đầu, mô hình phát triển phần mềm thác nước là xu hướng chủ đạo. Trong mô hình thác nước, việc thiết kế chiếm vai trò quan trọng và cần phải có trước khi phát triển. Với việc sử dụng mô hình thác nước, người ta sẽ bàn giao phần mềm trong cuối quá trình phát triển.

Với việc tập phân chia từng giai doạn cụ thể như vậy, việc xảy ra sai xót trong một giai đoạn đặc biệt là các giai đoạn ban đầu có thể khiến việc phát triển bị trì hoãn, tốn nhiều thời gian hơn dự kiến, phần mềm làm ra không thực sự thỏa mãn yêu cầu của khách hàng hoặc tệ hơn là dự án có thể bị thất bại hoàn toàn.

Trong bối cảnh đó, Agile ra đời quy định một tập các cách tiếp cận mới mẻ trong nhành công nghệ phần mềm nhằm khắc phục những điểm yếu của các phương pháp truyền thống. Phương pháp này đã được nhiều công ty trên thế giới áp dụng từ nhiều năm nay và tại Việt Nam nhiều công ty phần mềm cũng đã bắt đầu chú ý tới phương pháp này vì nó giúp quá trình phát triển phần mềm nhanh chóng, nhan có sản phẩm và tối ưu hóa đầu từ.

Agile là một tập các quy tắc mà các phương pháp phải tuân theo. Các phương pháp phát triển phần mềm như Scrum, XP (Extreme Programming), Feature Driven Development (FDD), Test Driven Development (TDD)... mới thực sự áp dụng các nguyên tắc này vào thực tế. Trong đó Scrum là quy trình đang được áp dụng rộng rãi nhất và mang lại hiệu quả rất cao (chiếm 52% trong tổng số các phương pháp – thống kê năm 2011).

Là một kĩ sư phần mềm việc tìm hiểu và nắm bắt Agile là một yêu cầu cần thiết vì vậy nhóm sinh viên đã thực hiện đề tài **“Nghiên cứu mô hình Agile và phát triển công cụ hỗ trợ”.**

**Mục đích**

Tìm hiểu và nắm bắt Agile và một số phương pháp, làm rõ ưu điểm và khuyết điểm của các chương trình đã có, từ đó xác định được mục tiêu phát triển chương trình của mình.

**Đối tượng**

**Phạm vi đề tài**

Phạm vi của đề tài là nghiên cứu cơ sở lý thuyết và nền tảng công nghệ để vận dụng cài đặt các tính năng cơ bản của một công cụ hỗ trợ quản lí dự án theo mô hình Scrum. Trong khuôn khổ luận văn, nhóm sẽ thực hiện một số vấn đề sau :

* Tìm hiểu, nghiên cứu Agile và phương pháp Scrum
* Xây dựng công cụ hỗ trợ quản lí dự án theo phương pháp Scrum.

# TỔNG QUAN

## Giới thiệu vấn đề

Hiện nay, có rất nhiều phương pháp/quy trình phát triển phần mềm, mỗi phương pháp có ưu nhược điểm riêng nhưng ta có thể chia làm hai loại chính: các phương pháp truyền thống và các phương pháp linh hoạt (Agile). Các dự án theo Agile dựa trên các giá trị thương mại, đáp ứng thích nghi với thay đổi thực tế hơn là theo kế hoạch đề ra như các phương pháp truyền thống.

Bên cạnh những lợi ích mà Agile mang lại, việc áp dụng nó cũng tồn tại một số khó khăn. Cụ thể, khi sử dụng quy trình Scrum, việc triển khai dự án có thể gặp một số khó khăn sau:

* Khó khăn về việc lưu trữ thông tin dự án đã làm trước đó nhằm thống kê, tham khảo hoặc phát triển thêm.
* Khó khăn về không gian khi yêu cầu tất cả các thành viên trong nhóm phải làm việc cùng nhau, việc này đôi khi là không thể.
* Trong thời buổi Internet phổ biến, việc sử dụng Agile thủ công không còn phù hợp. Người quản lí hoặc khách hàng không cần phải đền công ty mà có thể theo đõi tiến độ dự án một cách nhanh chóng và chình xác.
* Các thành viên trong nhóm có thể dễ dàng biết được tiến độ của các thành viên khác. Khi có vấn đề phát sinh thì các thành viên có thể hỗ trợ, giúp đỡ cùng giải quyết vấn đề.
* Một hệ thống quản lí sẽ có thêm những tính năng về báo cáo, thông kê, theo dõi tiến độ thuận tiện hơn nhiều so với việc sử dụng giấy bút truyền thống.

Với những khó khăn trên, thì việc cần có một công cụ hỗ trợ là rất cần thiết. Trên thị trường đã có một số công cụ thương mại nhưng các công này có một số hạn chế nhất định. Chi tiết kết quả khảo sát được trình bày trong mục 1.2

## Khảo sát một số công cụ có sẵn

|  |  |  |
| --- | --- | --- |
| Tên công cụ | Điểm hạn chế | Điểm cần khắc phục ở tool |
| **VersionOne**  www.versionone.com | * Là một tool rất mạnh hỗ trợ agile, Version one có tất cả những gì cần có cho một team scrum. Điểm yêu duy nhất của nó là quá phức tạp, quá nhiều tình năng không cần thiết và giao diện nhìn tương đối rắc rối. * Không có tính tương tác cao do vẫn đơn thuần là một website thông thường, người dùng muốn cập nhật thông tin phải Refresh lại trang. * Danh sách quản lí task thiếu tính trực quan, vẫn theo dạng bảng truyền thống, gây khó khăn cho người sử dụng. | * Giao diện nhìn trực quan hơn, đơn giản hơn và thuận tiện hơn cho người sử dụng. * Chỉ cần cung cấp các chức năng cần thiết với 1 tool Scrum. * Sử dụng những công nghệ mới cho web app, tăng cường khả năng tương tác giữa các thành viên trong team nhờ vào cơ chế tự động cập nhật khi có bất cứ thay đổi nào tới project. |
| *www.scrumwise.com* | * Tính năng Report còn sơ xài, thiếu xuất file báo cáo. * Vẫn theo mô hình website truyền thống nên thiếu tính cập nhật với dữ liệu. | * Tính năng report với nhiều loại biểu đồ, định dạng. |

## Mục tiêu xây dựng

Để hỗ trợ cho việc triển khai dự án theo Agile với phương pháp Scrum, đề tài sẽ tập trung vào nghiên cứu các nguyên tắc của Agile cùng với quy trình tiển khai Scrum. Từ những kiến thức đó, nhóm sẽ xây dựng một công cụ hỗ trợ quản lí dự án theo mô hình Scrum với một số chức năng cơ bản nhất.

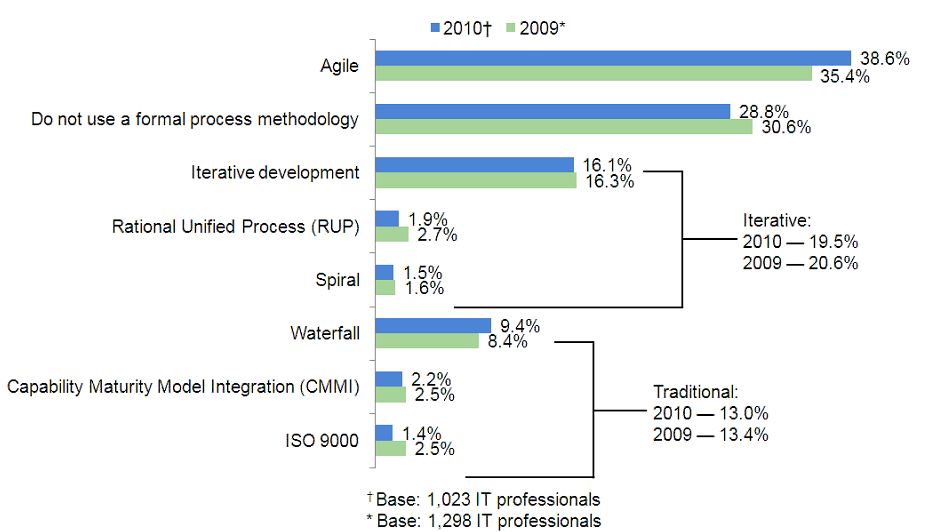
* + - Quản lí danh sách các dự án của đội ngũ phát triển (đã làm cũng như đang được tiến hành).
    - Hỗ trợ cho Product Owner quản lí danh sách các User Story, theo dõi được tiến độ dự án (thông qua tiến độ thực hiện của các Story).
    - Quản lí danh sách nhân lực của dự án cùng các Team phát triển.
    - Quản lí các thành phần của Scrum như Sprint, User Story, Task.
    - Cập nhật thời gian thực, người dùng sẽ không phải liên tục tải lại trang để cập nhật dữ liệu, công cụ sẽ liên tục tự động cập nhật mỗi khi dữ liệu thay đổi.
    - Biểu đồ thống kê thuận tiện cho người dùng theo dõi tiến độ của dự án.

# CƠ SỞ LÝ THUYẾT

## Giới thiệu về Agile

Phát triển phần mềm linh hoạt (Agile software development – gọi tắt là Agile) là một triết lí cùng với nhóm các phương pháp và phương pháp luận phát triển phần mềm dựa trên các nguyên tắc phát triển phân đoạn lặp (iterative) và tăng trưởng (incremental). Agile thường sử dụng cách lập kế hoạch thích ứng (adaptive planning), việc phát triển và chuyển giao theo hướng tiến hóa; sử dụng các khung thời gian ngắn và linh hoạt để dễ dàng phản hồi lại với các thay đổi trong quá trình phát triển. Ngày nay, triết lí Agile đã vượt xa khỏi khu vực truyền thống của mình là phát triển phần mềm mà còn đóng góp sự thay đổi trong cách thức làm việc, quản lí, sản xuất ở các ngành khác như sản xuất, dịch vụ, bán hàng, marketing, giáo dục...

Thuật ngữ "Agile" chính thức được sử dụng rộng rãi theo cách thống nhất kể từ khi "Tuyên ngôn Agile" được giới thiệu ra công chúng năm 2001. Nhờ tính linh hoạt, đa dạng và hiệu quả cao, các phương pháp Agile ngày càng trở thành sự lựa chọn hàng đầu của các khách hàng, nhà phát triển, các công ty phát triển phần mềm. Theo khảo sát của hãng nghiên cứu thị trường Forrester, mức độ phổ biến của Agile hiện đang ở mức cao nhất, và gấp nhiều lần so với các phương pháp truyền thống.



Hình .: Mức độ phổ biến của các phương pháp Agile

*(Nguồn: Forrester Research))*

### Tuyên ngôn Agile

Vào tháng 2 năm 2001, mười bảy nhà phát triển phần mềm đã gặp gỡ nhau ở Snowbird, Utah Resort để thảo luận về các phương pháp phát triển phần mềm gọn nhẹ và linh hoạt. Họ đã cùng nhau công bố "Tuyên ngôn Phát triển phần mềm linh hoạt" ("Manifesto for Agile Software Development" - gọi tắt là "Tuyên ngôn Agile") nhằm định nghĩa về Phát triển phần mềm linh hoạt. Đây là cột mốc quan trọng của agile cho dù trước đó, các phương pháp agile như XP, Scrum, v.v. đã được sử dụng thành công ở rất nhiều nơi, nhưng phải tới khi có sự xuất hiện của "Tuyên ngôn Agile", cùng với sự ra đời của các hiệp hội chuyên ngành agile như Agile Alliance hay Scrum Alliance, các phương pháp agile mới có một sự phát triển vượt bậc. Văn bản này rất ngắn, dễ hiểu, và rất quan trọng vì nó đưa ra các giá trị cốt lõi nhất mà toàn bộ những người dùng Agile tuân thủ; mặc dù các phương pháp họ đề xuất hoặc sử dụng trong thực tiễn có thể rất khác nhau. Toàn văn Tuyên ngôn Agile như sau:

*Chúng tôi đã phát hiện ra cách phát triển phần mềm tốt hơn bằng cách hiện thực nó và giúp đỡ người khác thực hiện. Qua quá trình này, chúng tôi đã đi đến việc đánh giá cao:*

* ***Cá nhân và sự tương tác****hơn là quy trình và công cụ.*
* ***Phần mềm chạy tốt****hơn là tài liệu đầy đủ.*
* ***Cộng tác với khách hàng****hơn là đàm phán hợp đồng.*
* ***Phản hồi với các thay đổi****hơn là bám sát kế hoạch.*

*Mặc dù các điều bên phải vẫn còn giá trị, nhưng chúng tôi đánh giá cao hơn các mục ở bên trái.*

Bên cạnh đó, các nhà phát triển còn nhấn mạnh mười hai nguyên lý sau Tuyên ngôn Agile nhằm giúp các nhà phát triển vận dụng các phương pháp agile trong thực tiễn. Các nguyên lý được liệt kê sau đây:

* Ưu tiên cao nhất của chúng tôi là thỏa mãn khách hàng thông qua việc chuyển giao sớm và liên tục các phần mềm có giá trị.
* Chào đón việc thay đổi yêu cầu, thậm chí rất muộn trong quá trình phát triển. Quy trình Agile thực hiện những thay đổi phục vụ cho lợi thế cạnh tranh của khách hàng.
* Thường xuyên chuyển giao phần mềm tới khách hàng, từ vài tuần đến vài tháng, ưu tiên cho các khoảng thời gian ngắn hơn.
* Nhà kinh doanh và nhà phát triển phải làm việc cùng nhau hàng ngày trong suốt dự án.
* Xây dựng các dự án xung quanh những cá nhân tận tụy. Cung cấp cho họ môi trường và sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc.
* Phương pháp hiệu quả nhất để truyền đạt thông tin tới và trong nội bộ nhóm phát triển là đối thoại trực tiếp.
* Phần mềm chạy tốt là thước đo chính của tiến độ.
* Các quy trình Agile thúc đẩy sự phát triển bền vững. Các nhà tài trợ, nhà phát triển, và người dùng có thể duy trì một nhịp độ liên tục không giới hạn.
* Liên tục quan tâm đến các kĩ thuật nổi trội và thiết kế tốt để gia tăng sự linh hoạt.
* Sự đơn giản – nghệ thuật tối đa hóa lượng công việc chưa xong – là căn bản.
* Các kiến trúc tốt nhất, yêu cầu tốt nhất, và thiết kế tốt nhất sẽ được làm ra bởi các nhóm tự tổ chức.
* Đội ngũ sẽ thường xuyên suy nghĩ về việc làm sao để trở nên hiệu quả hơn, sau đó họ sẽ điều chỉnh và thay đổi các hành vi của mình cho phù hợp.

Các nguyên lý này, cùng với năm điểm cốt lõi trong "Tuyên ngôn Agile" sẽ dẫn đường cho các nhà thực hành agile (agile practictioner) vận dụng tốt các phương pháp agile vào thực tiễn.

### Giá trị cốt lõi của Agile

#### Cá nhân và sự tương tác

Cá nhân và sự tương tác giữa họ là nhân tố chủ yếu để nhóm đạt được hiệu suất cao. Các nghiên cứu về "bão hòa truyền thông" trong một dự án cho thấy rằng, khi không có vấn đề trong giao tiếp, nhóm có thể thực hiện công việc tốt hơn 50 lần so với mức trung bình trong lĩnh vực của mình. Để tạo điều kiện cho việc giao tiếp, các phương pháp linh hoạt thường xuyên sử dụng chu trình thanh-tra-và-thích-nghi (Inspect and Adapt). Chu trình này có thể diễn ra hằng phút với lập trình cặp (pair-programming), hằng giờ với tích hợp liên tục (continuous integration), hằng ngày với standup metting (đứng họp), hằng phân đoạn với các buổi họp sơ kết và cải tiến.

Tuy nhiên, chỉ tăng tần suất phản hồi và giao tiếp là không đủ để loại bỏ các vấn đề về truyền thông. Chu kỳ thanh-tra-và-thích-nghi làm việc tốt chỉ khi các thành viên trong nhóm thể hiện những hành vi quan trọng sau:

* Tôn trọng giá trị của mỗi cá nhân.
* Trung thực trong truyền thông.
* Minh bạch về dữ liệu, hoạt động, và quyết định.
* Tin tưởng vào sự hỗ trợ của mỗi cá nhân với nhóm.
* Cam kết với nhóm và các mục tiêu của nhóm.

Để thúc đẩy các hành vi này, nhà quản lý linh hoạt phải cung cấp một môi trường hỗ trợ, các nhà huấn luyện phải tạo điều kiện thuận lợi, và các thành viên của nhóm phải thể hiện chúng. Chỉ khi đó nhóm mới có thể phát huy được hết tiềm năng của mình. Đạt tới các hành vi đó khó khăn hơn rất nhiều so với bề ngoài của chúng. Hầu hết các nhóm tránh lé sự thật, sự minh bạch, và tin tưởng vào các chuẩn mực văn hóa hoặc có những kinh nghiệm tiêu cực từ các xung đột. Để chống lại những khuynh hướng này, ban lãnh đạo và thành viên nhóm phải tạo điều kiện cho những xung đột tích cực nhằm mục đích:

* Cải tiến quy trình phụ thuộc vào nhóm trong việc tạo ra danh sách các trở ngại hoặc vấn đề trong tổ chức, đối mặt với chúng một cách trung thực, và sau đó loại bỏ chúng một cách có hệ thống tùy theo độ ưu tiên.
* Đổi mới chỉ xảy ra với việc tự do trao đổi các ý tưởng mâu thuẫn lẫn nhau, một hiện tượng được nghiên cứu và viết thành tài liệu bởi Takeuchi và Nonaka, những người cha đỡ đầu của Scrum.
* Việc hướng các nhóm tới mục tiêu chung đòi hỏi nhóm phải đối mặt và giải quyết các vấn đề về xung đột.
* Cam kết làm việc cùng nhau sẽ xảy ra chỉ khi mọi người đồng ý với mục tiêu chung và sau đó đấu tranh cho sự tiến bộ của bản thân cũng như nhóm.

Ý cuối cùng ở trên nói về cam kết là đặc biệt quan trọng. Đó là khi mà các cá nhân và các nhóm được cam kết mà họ cảm thấy có trách nhiệm với việc cung cấp các giá trị cao, đó là điểm mấu chốt đối với các nhóm phát triển phần mềm. Các phương pháp linh hoạt tạo điều kiện cho việc cam kết bằng cách khuyến khích các nhóm đưa ra một danh sách các công việc được ưu tiên hóa, để họ tự quản lý công việc của mình, và tập trung vào cải tiến về cách thực hiện các công việc đó. Thực hành là nền tảng của tự-tổ-chức (self-organization), đó là động lực để đạt được kết quả trong một nhóm agile.

Để tạo ra các nhóm có hiệu suất cao, các phương pháp linh hoạt coi trọng cá nhân và sự tương tác hơn là quy trình và công cụ. Thực tế cho thấy rằng, tất cả các phương pháp linh hoạt tìm kiếm sự gia tăng trong truyền thông và cộng tác thông qua việc kiểm tra thường xuyên các chu trình thanh-tra-và-thích-nghi. Tuy nhiên, các chu trình đó chỉ thực sự làm việc tốt khi mà các nhà lãnh đạo agile khuyến khích các cuộc xung đột tích cực, thứ cần thiết để xây dựng một nền tảng vững chắc cho sự trung thực, tính minh bạch, lòng tin, sự tôn trọng, và cam kết từ các nhóm agile của họ.

#### Phần mềm chạy tốt hơn là tài liệu đầy đủ

Phần mềm chạy tốt là một trong những khác biệt lớn mà phát triển linh hoạt mang lại. Tất cả các phương pháp linh hoạt thể hiện Tuyên ngôn Agile bằng cách nhấn mạnh việc cung cấp một phần nhỏ của phần mềm chạy tốt tới khách hàng sau một khoảng thời gian nhất định.

#### Cộng tác với khách hàng hơn là thương thảo hợp đồng

Trong hai thập kỷ qua, tỉ lệ thành công của các dự án tăng hơn hai lần trên toàn thế giới. Điều này được cho là vì các dự án nhỏ hơn và mức độ chuyển giao thường xuyên đã cho phép khách hàng cung cấp các thông tin phản hồi về phần mềm một cách đều đặn hơn. Các tác giả của bản tuyên ngôn đã làm sáng tỏ điều này khi họ nhấn mạnh rằng việc để khách hàng tham gia vào quá trình phát triển phần mềm là hết sức cần thiết để dẫn tới việc thành công cho dự án.

Các phương pháp phát triển linh hoạt đã thúc đẩy giá trị này bằng cách đưa vào một đồng minh tích cực của khách hàng làm việc sát cánh với đội phát triển. Lấy một ví dụ, một nhóm Scrum đầu tiên có hàng ngàn khách hàng. Sẽ là không khả thi nếu cho phép tất cả khách hàng tham gia vào quá trình phát triển sản phẩm, vì vậy họ chọn ra một vị đại sứ của khách hàng, được gọi là Product Owner, để đại diện cho không chỉ tất cả khách hàng trong trường hợp này mà còn bao gồm cả quản lý, dịch vụ khách hàng, và các bên liên quan khác. Product Owner có trách nhiệm cập nhật danh sách yêu cầu về sản phẩm sau mỗi bốn tuần thời điểm mà nhóm Scrum phát hành phiên bản sản phẩm chạy tốt, có tính đến yếu tố thực tế cùng phản hồi của khách hàng và các bên liên quan. Điều này cho phép khách hàng có thể giúp định hình sản phẩm phần mềm đang được tạo ra.

Cộng tác với khách hàng (hoặc đại diện của khách hàng) trên cơ sở hàng ngày là một trong những lý do lý giải tại sao các dự án Agile có tỉ lệ thành công cao hơn gấp đôi so với các dự án truyền thống tính trung bình trên toàn thế giới. Các phương pháp phát tuân theo Agile đểu tạo ra một vị trí đặc biệt trong đội ngũ phát triển để dành riêng cho vị khách hàng đại diện này.

#### Phản hồi với thay đổi hơn là bám sát kế hoạch

Phản hồi với thay đổi là điều cần thiết cho việc tạo ra một sản phẩm làm hài lòng khách hàng cũng như mang lại những giá trị kinh doanh. Dữ liệu ngành công nghiệp cho thấy hơn 60% các yêu cầu về sản phẩm hay dự án bị thay đổi suốt quá trình phát triển phần mềm. Ngay cả khi các dự án truyền thống kết thúc đúng thời gian, đúng kinh phí, với tất cả các tính năng theo kế hoạch, nhưng khách hàng thường không hài lòng vì những gì họ thấy không thật sự đúng như những gì họ muốn. Luật Humphrey nói rằng khách hàng không bao giờ biết những gì họ muốn cho đến khi họ thấy phần mềm hoạt động. Nếu khách hàng không nhìn thấy phần mềm hoạt động cho đến khi kết thúc dự án, sẽ là quá muộn cho việc kết hợp các thông tin phản hồi của họ ở thời điểm này. Các phương pháp phát triển linh hoạt tìm kiếm sự phản hồi của khách hàng trong suốt dự án để có thể kết hợp thông tin phản hồi và thông tin mới ngay cả khi sản phẩm đang được phát triển.

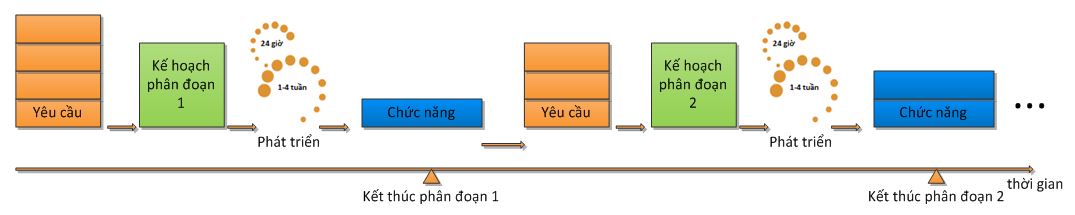
Tất cả các phương pháp phát triển linh hoạt đều được tích hợp sẵn những quy trình với các thay đổi trong một khoảng thời gian đều đặn dựa trên những thông tin phản hồi từ phía khách hàng cũng như bên đại diện của khách hàng. Các kế hoạch được thiết kế để sao cho luôn cung cấp giá trị kinh doanh cao nhất trước hết. Bởi vì 80% giá trị nằm trong 20% các tính năng, một dự án phát triển linh hoạt chạy tốt có xu hướng kết thúc sớm, trong khi hầu hết các dự án truyền thống thường kết thúc trễ.

### Đặc trưng của Agile

Có rất nhiều phương pháp agile với các hướng tiếp cận rất khác nhau. Bên cạnh các cách thức tổ chức công việc, thiết lập quy trình, các phương pháp agile còn nghiên cứu và đưa vào sử dụng các công cụ và kĩ thuật đặc thù như công cụ tích hợp liên tục (continuous integration), kiểm thử đơn vị, mẫu thiết kế, tái cấu trúc, phát triển hướng kiểm thử (Test-Driven Development), phát triển hướng hành vi (Behaviour-Driven Development), hay lập trình theo cặp (Pair Programming)... để đảm bảo và gia tăng tính linh hoạt. Tuy vậy các phương pháp này chia sẻ nhiều đặc trưng giống nhau: cộng tác nhóm chặt chẽ, tổ chức các nhóm tự quản, liên chức năng, tính đáp ứng cao trong suốt vòng đời của dự án.

#### Tính lặp

Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại. Các phân đoạn (được gọi là Iteration hoặc Sprint) này thường có khung thời gian ngắn (từ một đến bốn tuần). Trong mỗi phân đoạn này, nhóm phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử (với các mức độ khác nhau) để cho ra các phần nhỏ của sản phẩm. Các phương pháp Agile thường phân rã dự án thành các phần nhỏ với quá trình lập kế hoạch đơn giản và gọn nhẹ nhất có thể, và không thực hiện việc lập kế hoạch dài hạn. Khi đó, thậm chí công việc lập kế hoạch, làm mịn kế hoạch được thực hiện liên tục trong suốt quá trình làm việc.



Hình .: Các phân đoạn lặp trong Agile

*(Nguồn: hanoiscrum.net)*

#### Tính tăng và tiến hóa

Cuối các phân đoạn, nhóm phát triển thường cho ra các phần nhỏ của sản phẩm cuối cùng. Các phần nhỏ này thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng ngay. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn. Khác với mô hình phát triển thác nước – vốn chỉ cho phép nhìn thấy toàn bộ các chức năng tại thời điểm kết thúc dự án, sản phẩm trong các dự án Agile lớn dần lên theo thời gian, tiến hóa cho tới khi đạt được trạng thái đủ để phát hành.

#### Tính thích ứng

Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn, và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu ...) đều có thể được đáp ứng theo cách thích hợp. Ví dụ, trong Scrum – phương pháp phổ biến nhất hiện nay – trong khi nhóm phát triển sản xuất ra các gói phần mềm, khách hàng có thể đưa thêm các yêu cầu mới, Product Owner có thể đánh giá các yêu cầu này và có thể đưa vào làm việc trong phân đoạn (được gọi là Sprint trong Scrum) tiếp theo. Nhờ đó, các quy trình Agile thường thích ứng rất tốt với các thay đổi.

#### Nhóm tự tổ chức và liên chức năng

Cấu trúc nhóm agile thường là liên chức năng và tự tổ chức. Theo đó, các nhóm này tự thực hiện lấy việc phân công công việc mà không dựa trên các mô tả cứng nhắc về chức danh hay làm việc dựa trên một sự phân cấp rõ ràng trong tổ chức. Các nhóm này cộng tác với nhau để ra quyết định, theo dõi tiến độ, giải quyết các vấn đề mà không chờ mệnh lệnh của các cấp quản lý. Họ không làm việc theo cơ chế "mệnh lệnh và kiểm soát". Nhóm tự tổ chức có nghĩa là nó đã đủ các kĩ năng cần thiết cho việc phát triển phần mềm, do vậy nó có thể được trao quyền để tự ra quyết định, tự quản lí và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

#### Quản lí tiến trình thực nghiệm

Các nhóm agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay đưa ra các giả định. Việc phân nhỏ dự án thành các phân đoạn ngắn góp phần gia tăng các điểm mốc để nhóm phát triển thu thập dữ kiện cho phép điều chỉnh các chiến lược phát triển của mình. Nói cách khác, Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt. Theo thời gian, các chiến lược này sẽ tiến gần đến trạng thái tối ưu, nhờ đó nhóm có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

#### Đối thoại trực diện

Một số mô hình phát triển phần mềm dựa rất nhiều vào công việc giấy tờ, từ việc thu thập yêu cầu người dùng, viết đặc tả hệ thống, các thiết kế hệ thống v.v. Agile không phản đối công dụng của công việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì gián tiếp thông qua giấy tờ. Về yêu cầu của khách hàng, agile khuyến khích nhóm phát triển trực tiếp nói chuyện với khách hàng để hiểu rõ hơn về cái khách hàng thực sự cần, thay vì phụ thuộc nhiều vào các loại văn bản. Trong giao tiếp giữa nội bộ nhóm phát triển với nhau, thay vì một lập trình viên (thực hiện việc mã hóa) và một kĩ sư (thực hiện việc thiết kế) giao tiếp với nhau thông qua bản thiết kế, agile khuyến khích hai người này trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

Bản thân các nhóm agile thường nhỏ (ít hơn mười hai người, một nhóm lớn thường được phân nhỏ với cơ chế hợp tác đặc thù để luôn luôn có thông lượng giao tiếp tối đa) để đơn giản hóa quá trình giao tiếp, thúc đẩy việc cộng tác hiệu quả. Các dự án lớn muốn dùng agile thường phải phân thành nhóm nhỏ đồng thời làm việc với nhau hướng đến một mục tiêu chung. Việc này có thể đòi hỏi nhiều nỗ lực trong việc điều phối các mức ưu tiên giữa các nhóm.

Các nhóm phát triển thường tạo ra các thói quen và cơ chế trao đổi trực diện thường xuyên. Một trong các cơ chế thường thấy là các cuộc họp tập trung hằng ngày (daily meeting, daily Scrum, standup meeting). Tại đây, tất cả các thành viên được yêu cầu nói rõ cho nhóm của mình biết mình đã làm gì, đang làm gì, sắp làm gì và đang gặp phải khó khăn nào trong quá trình làm việc. Khi cơ chế này được thực hiện hiệu quả, nhóm luôn luôn nắm được tình hình công việc của mình, có các hành động thích hợp để vượt qua các cản trở để thực hiện thành công mục tiêu của dự án.

#### Phát triễn dựa trên giá trị

Một trong các nguyên tắc cơ bản của agile là "phần mềm chạy tốt chính là thước đo của tiến độ". Nguyên tắc này giúp nhóm dám loại bỏ đi các công việc dư thừa không trực tiếp mang lại giá trị cho sản phẩm. Ví dụ, một nhóm thấy rằng có thể không cần phải viết ra tất cả các thiết kế của hệ thống, mà họ chỉ cần phác thảo các thiết kế này lên bảng, rồi cùng nhau triển khai các chức năng của hệ thống. Nhưng nếu như Product Owner quyết định rằng, khi chuyển giao sản phẩm, nhóm phát triển phải kèm theo thiết kế chi tiết của hệ thống vì chúng chiếm 20% giá trị của dự án theo yêu cầu của khách hàng, và vì khách hàng cần nó để chứng minh tính đúng đắn của các chức năng, và để phát triển tiếp các phân hệ tiếp theo của sản phẩm; thì nhóm phát triển sẽ phải thực hiện việc tài liệu hóa đầy đủ.

Để vận hành được cơ chế "làm việc dựa trên giá trị", nhóm agile thường làm việc trực tiếp và thường xuyên với khách hàng (hay đại diện của khách hàng), cộng tác trực tiếp với họ để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án. Nhờ đó các dự án agile thường giúp khách hàng tối ưu hóa được giá trị của dự án. Vì thế, agile gia tăng đáng kể độ hài lòng của khách hàng.

## Scrum

### Giới thiệu Scrum

Scrum là một khung làm việc trong đó con người có thể xác định các vấn đề thích nghi phức tạp, trong khi vẫn giữ được năng suất và sáng tạo để chuyển giao các sản phẩm có giá trị cao nhất. Scrum có các tính chất:

* Nhẹ nhàng
* Dễ hiểu
* Rất khó để tinh thông

Scrum là khung làm việc đã được sử dụng để quản lý quá trình phát triển các sản phẩm phức tạp từ đầu những năm 1990. Scrum không phải là một quy trình hay một kĩ thuật cụ thể để xây dựng sản phẩm; hơn thế, nó là một khung làm việc cho phép bạn sử dụng nhiều quy trình và kĩ thuật khác nhau.

Khung làm việc Scrum bao gồm một nhóm Scrum với các vai trò được phân định rõ ràng, các sự kiện, các đồ nghề và các quy tắc. Mỗi thành phần trong khung làm việc phục vụ một mục đích rõ ràng và có vai trò nòng cốt trong cho thành công của Scrum.

### Lý thuyết Scrum

Scrum được xây dựng dựa trên lý thuyết quản lý tiến trình thực nghiệm. Lý thuyết này chỉ ra rằng tri thức đến từ kinh nghiệm và việc ra quyết định được dựa trên những gì đã biết. Scrum sử dụng các tiếp cận lặp, tăng dần để tối ưu hóa việc dự đoán và kiểm soát rủi ro.

Ba yếu tố nòng cốt tạo thành một mô hình quản lý tiến trình thực nghiệm gồm: tính minh bạch (transparency), sự thanh tra (inspection) và sự thích nghi (adaptation).

**Minh bạch**

Các khía cạnh quan trọng của tiến trình phải được hiển thị rõ ràng cho những người có nhiệm với thành quả của tiến trình đó. Sự minh bạch yêu cầu các yếu tố này định nghĩa theo một tiêu chuẩn để những người quan sát có thể hiểu những gì họ thấy theo cùng một cách.

**Thanh tra**

Người sử dụng Scrum phải thường xuyên thanh tra các đồ nghề và tiến độ đến đích để phát hiện các bất thường không theo ý muốn. Tần suất thanh tra không nên quá dày để khỏi ảnh hưởng đến công việc. Công tác thanh tra có ích nhất khi được thực hiện bởi người có kĩ năng tại các điểm quan trọng của công việc.

**Thích nghi**

Nếu một người thanh tra xác định được rằng có vấn đề nào đó vượt quá giới hạn cho phép, và hậu quả của vấn đề đó đối với sản phẩm là không thể chấp nhận được, thì quy trình hoặc phải được điều chỉnh. Sự điều chỉnh phải được tiến hành càng sớm càng tốt để giảm thiểu các sai sót khác có thể xảy ra.

Scrum cung cấp bốn cơ hội chính thức cho việc thanh tra và thích nghi trong các Sự kiện Scrum, bao gồm:

* Buổi họp kế hoạch Sprint (Sprint Planning Meeting)
* Họp Scrum hằng ngày (Daily Scrum)
* Sơ kết Sprint (Sprint Review)
* Cải tiến Sprint (Sprint Retrospective)

### Nhóm Scrum

Nhóm Scrum bao gồm Product Owner, Team, và Scrum Master. Các Nhóm Scrum là các nhóm tự quản và liên chức năng. Các nhóm tự quản tự mình chọn cách thức tốt nhất để hoàn thành công việc của họ, chứ không bị chỉ đạo bởi ai đó bên ngoài nhóm. Các nhóm liên chức năng có đủ kĩ năng cần thiết để hoàn thành công việc mà không phụ thuộc vào bất kì người ngoài nào khác. Mô hình nhóm trong Scrum được thiết kế để tối ưu hóa sự linh hoạt, sáng tạo và năng suất.

Các Nhóm Scrum chuyển giao sản phẩm theo phân đoạn lặp đi lặp lại và tăng dần, tối đa hóa cơ hội cho các phản hồi. Việc chuyển giao tăng dần các gói sản phẩm đạt tiêu chuẩn “hoàn chỉnh” đảm bảo một phiên bản khả dụng của sản phẩm luôn luôn sẵn sàng để cung cấp tới Product Owner.

**Product Owner**

Product Owner chịu trách nhiệm tối đa hóa giá trị của sản phẩm và công việc của nhóm phát triển. Cách thức để đạt được điều đó có thể rất khác nhau giữa các tổ chức, nhóm Scrum và các cá nhân.

Product Owner là một người chủ yếu chịu trách nhiệm về việc quản lý Product Backlog:

* Mô tả các hạng mục Product backlog;
* Trình tự của các hạng mục trong Product Backlog để đạt được mục đích và hoàn thành dự án.
* Đảm bảo cho Product Backlog là luôn luôn thông suốt, và rõ ràng tới tất cả mọi người, và chỉ ra những gì mà Nhóm Scrum sẽ làm.
* Đảm bảo cho Team hiểu rõ các hạng mục trong Product Backlog với các mức độ cần thiết.

Product Owner có thể thực hiện công việc trên, hoặc để Team làm. Tuy nhiên, Product Owner vẫn phải chịu trách nhiệm chính. Product Owner là một người, không phải là một ủy ban. Product Owner có thể cần tới một ủy ban tham gia vào Product Backlog, nhưng những người trong ủy ban muốn thay đổi trình tự các hạng mục trong Product Backlog phải thuyết phục được Product Owner.

Để Product Owner thành công, cả tổ chức phải tôn trọng các quyết định của người này. Các quyết định đó được hiển thị trong nội dung và thứ tự trong Product Backlog. Không ai ngoài Product Owner được phép yêu cầu Team làm gì khác, và Team cũng không được phép làm gì theo lời bất cứ người nào khác.

**Team**

Team (nhóm phát triển) gồm các chuyên gia làm việc để cho ra các phần tăng trưởng có thể phát hành được vào cuối mỗi Sprint. Chỉ các thành viên của Team mới tạo ra các phần tăng trưởng này. Team được cấu trúc và trao quyền để tổ chức và quản lý công việc của họ. Team có các đặc trưng sau:

* Đó là nhóm tự tổ chức. Không ai (kể cả Scrum Master) có quyền yêu cầu Team làm sao để chuyển Product Backlog thành các phần tăng trưởng có thể chuyển giao được.
* Đó là nhóm liên chức năng, với tất cả các kĩ năng cần thiết để tạo ra phần tăng trưởng của sản phẩm.
* Scrum không ghi nhận một chức danh nào trong Team ngoài Developer, theo tính chất công việc của người này; không có ngoại lệ cho quy tắc này.
* Các thành viên Team có thể có các kĩ năng chuyên biệt và các chuyên môn đặc thù, nhưng họ phải chịu trách nhiệm dưới một thể thống nhất là Team.
* Team không chứa các nhóm con nào khác với các chức năng đặc thù như ‘nhóm kiểm thử’ hay ‘phân tích nghiệp vụ’.

Độ lớn tối ưu của Team là đủ nhỏ để giữ được sự linh hoạt và đủ lớn để hoàn thành công việc. Ít hơn ba người có thể làm giảm sự tương tác và dẫn đến năng suất thấp. Các Team nhỏ hơn có thể phải đối mặt với các ràng buộc kĩ năng trong suốt Sprint, dẫn đến Team khó có thể chuyển giao gói tăng trưởng phát hành được. Một nhóm nhiều hơn chín người cần nhiều sự điều phối hơn. Các Team lớn phát sinh quá nhiều phức tạp để thực hiện việc kiểm soát tiến trình thực nghiệm. Các vai trò Product Owner và Scrum Master không được tính vào kích thước của Team, trừ khi họ cũng kiêm luôn vai trò là thành viên của Team.

**Scrum Master**

Scrum Master chịu trách nhiệm đảm bảo mọi người hiểu và dùng được Scrum. Scrum Master thực hiện việc này bằng cách đảm bảo Nhóm Scrum tuân thủ lý thuyết, thực tiễn và các quy tắc của Scrum. Scrum Master là một lãnh đạo, nhưng cũng là đầy tớ của nhóm Scrum.

Scrum Master giúp đỡ những người ngoài nhóm Scrum hiểu cách phải tương tác với nhóm sao cho hiệu quả nhất. Scrum Master giúp đỡ tất cả mọi người thay đổi các mối tương tác này để tối đa hóa giá trị mà nhóm Scrum tạo ra.

### Các sự kiện Scrum

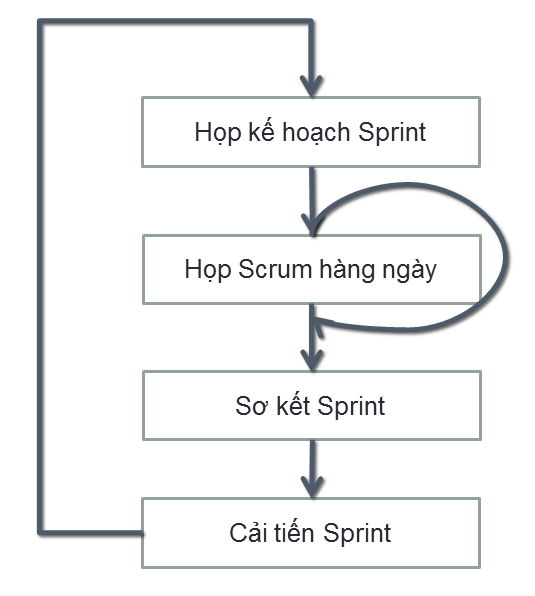
Các sự kiện được mô tả trong Scrum nhằm tạo ra thói quen và để giảm thiểu những buổi họp hành vốn không được định nghĩa trong Scrum. Scrum dùng các sự kiện được đóng khung thời gian (time-boxed), nghĩa là mỗi sự kiện có giới hạn thời gian tối đa. Điều này đảm bảo thời lượng vừa đủ để tránh lãng phí thời gian không cần thiết cho sự kiện.

Bao gồm cả bản thân Sprint vốn chứa tất cả các sự kiện khác, mỗi sự kiện trong Scrum là một cơ hội chính thức để thực hiện cơ chế thanh tra và thích nghi. Các sự kiện này được thiết kế đặc biệt để đảm bảo sự minh bạch và thanh tra. Nếu không thực hiện được các điều này có thể dẫn đến giảm thiểu tính minh bạch và đánh mất cơ hội để thanh tra và thích nghi.

**Sprint**

Trái tim của Scrum chính là Sprint, một khung-thời-gian (time-box) có thời gian một tháng hoặc ngắn hơn để tạo ra các phần tăng trưởng của sản phẩm có thể phát hành được. Sprint có khoảng thời gian nhất quán trong suốt quá trình phát triển. Một Sprint mới bắt đầu ngay khi Sprint trước khép lại.

Sprint bao gồm một cuộc họp kế hoạch Sprint (Sprint Planning Meeting), các cuộc họp Scrum hằng ngày (Daily Scrum), một buổi sơ kết Sprint (Sprint Review), và một buổi họp cải tiến Sprint (Sprint Retrospective).



Hình .: Các sự kiện Sprint

Trong suốt Sprint:

* Không cho phép bất kì sự thay đổi nào ảnh hưởng đến mục tiêu của Sprint.
* Thành phần Team được giữ nguyên.
* Mục tiêu chất lượng không được cắt giảm.
* Phạm vi dự án có thể được làm rõ và tái thương lượng giữa Product Owner và Team.

Mỗi Sprint có thể được coi như một tiểu dự án với độ dài một tháng. Giống như dự án, Sprint được dùng để hoàn tất điều gì đó. Mỗi Sprint có một định nghĩa về việc phải xây dựng cái gì, một bản thiết kế và bản kế hoạch linh hoạt sẽ hướng dẫn quá trình xây dựng đó, các công việc cần làm, và sản phẩm của quá trình đó. Sprint được giới hạn trong vòng một tháng lịch (calendar month). Khi Sprint bị kéo dài quá thì định nghĩa về việc phải xây dựng cái gì có thể bị thay đổi, sự phức tạp sẽ gia tăng và rủi ro sẽ tăng theo. Sprint đảm bảo tính dự đoán bằng sự thanh tra và thích nghi trong tiến trình tiến tới mục tiêu của mỗi tháng đó. Sprint cũng sẽ giới hạn rủi ro trong phạm vi chi phí của một tháng.

Sprint có thể bị hủy trước khi khung thời gian trôi qua. Chỉ có Product Owner mới đủ thẩm quyền kết thúc Sprint, mặc dù Product Owner có thể chịu ảnh hưởng bởi những bên hữu quan khác, bởi Team, hoặc bởi Scrum Master. Một Sprint có thể bị hủy nếu như mục tiêu Sprint có thể trở nên lỗi thời. Điều này xảy ra khi công ty chuyển hướng kinh doanh hoặc khi tình thế công nghệ có sự thay đổi. Nói chung, Sprint có thể bị hủy nếu nó không mang lại điều gì có ích. Thế nhưng, do thời gian trong mỗi Sprint tương đối ngắn, nên việc hủy một Sprint không mấy khi gây hậu quả lớn.

Khi Sprint bị hủy, các phần sản phẩm đã hoàn chỉnh được xem xét lại. Nếu phần nào đó của công việc đó là có thể chuyển giao được, thì Product Owner có thể chấp nhận chúng. Tất các các hạng mục Product Backlog chưa hoàn tất sẽ được ước lượng lại và đặt ngược trở lại Product Backlog để phát triển tiếp. Việc hủy Sprint sẽ gây lãng phí tài nguyên, do mọi người phải họp lại để lên kế hoạch cho một Sprint mới. Việc hủy Sprint thường gây tổn hại nhất định cho Team, và rất ít khi xảy ra.

**Họp kế hoạch Sprint**

Công việc trong Sprint được lên kế hoạch trong buổi họp kế hoạch Sprint (Sprint Planning Meeting). Kế hoạch cho Sprint được tạo ra nhờ nỗ lực cộng tác của toàn bộ Nhóm Scrum.

Buổi họp kế hoạch Sprint được đóng khung trong tám tiếng cho mỗi Sprint một tháng. Với các Sprint ngắn hơn thì thời gian cho buổi họp được rút ngắn lại. Ví dụ như một Sprint hai tuần có thể chỉ cần họp kế hoạch tới bốn tiếng là đủ.

Buổi họp kế hoạch Sprint có hai phần, mỗi phần chiếm một nửa khung thời gian. Hai phần của buổi họp kế hoạch Sprint lần lượt trả lời hai câu hỏi sau đây:

* Sprint này phải chuyển giao cái gì?
* Làm sao để đạt được điều đó?

**Họp Scrum hàng ngày**

Cuộc họp Scrum Hằng (Daily Scrum) ngày được đóng khung trong 15 phút để Team đồng bộ hóa các hoạt động của thành viên và tạo lập kế hoạch cho 24 giờ tiếp theo. Điều này có được nhờ việc thanh tra các công việc kể từ cuộc họp Scrum Hằng ngày trước, và dự báo những công việc sẽ được hoàn thành trước buổi họp lần sau.

Cuộc họp Scrum hằng ngày được tổ chức tại cùng một địa điểm để giảm thiểu sự phức tạp không cần thiết. Trong suốt cuộc họp, mỗi thành viên Team giải thích rõ:

* Việc gì đã được thực hiện kể từ lần họp trước?
* Việc gì sẽ được hoàn thành trước buổi họp lần sau?
* Có vấn đề gì nảy sinh trong quá trình làm việc?

Team sử dụng cuộc họp Scrum hằng ngày để đánh giá tiến độ công việc hướng tới mục tiêu Sprint và đánh giá xu hướng tiến triển của công việc trong Sprint Backlog. Cuộc họp Scrum hằng ngày tối ưu hóa khả năng để Team có thể đạt được mục tiêu của Sprint. Team thường họp mặt ngay sau khi họp xong Scrum hằng ngày để tái lập kế hoạch cho các công việc còn lại trong Sprint. Hằng ngày, Team có thể giải thích cho Product Owner và Scrum Master biết họ định làm gì với tư cách là một nhóm tự quản để hoàn thành mục tiêu và tạo ra các phần tăng trưởng cần thiết trong Sprint.

Scrum Master đảm bảo cho Team tham gia họp, nhưng chính Team mới có trách nhiệm chính trong cuộc họp Scrum hằng ngày. Scrum Master phải đảm bảo Team giữ cuộc họp ngắn gọn trong phạm vi khung thời gian 15 phút.

Scrum Master phải áp đặt quy tắc về việc chỉ có Team mới được tham gia cuộc họp Scrum hằng ngày. Cuộc họp Scrum hằng ngày không phải là cuộc họp báo cáo tình hình, mà là để mọi người biến đổi các hạng mục Product Backlog thành ra phần tăng trưởng cần thiết.

Họp Scrum hằng ngày sẽ cải tiến quá trình giao tiếp, lược bỏ các buổi họp hành không cần thiết, nhận biết và loại bỏ các trở lực trong quá trình phát triển, nhấn mạnh và phát huy các quyết định nhanh chóng, và nâng cao mức độ hiểu biết của Team về dự án. Cuộc họp này là chìa khóa của cơ chế thanh tra và thích nghi trong Scrum.

**Sơ kết Sprint**

Buổi Sơ kết Sprint (Sprint Review) được tổ chức khi Sprint kết thúc để rà soát lại phần tăng trưởng vừa làm ra trong Sprint đó, và để thực hiện các biện pháp thích nghi nếu cần. Trong cuộc họp này, Nhóm Scrum và các bên hữu quan sẽ trao đổi với nhau về những gì vừa hoàn thành trong Sprint vừa rồi. Trên cơ sở đó và những sự thay đổi trong Product Backlog trong suốt quá trình thực hiện Sprint, người tham dự cuộc họp sẽ hợp tác để thảo luận về những công việc sắp triển khai. Đây là cuộc họp không trang trọng, và việc trình bày về gói tăng trưởng chủ yếu nhằm mục đích cung cấp các phản hồi hữu ích và khuyến khích sự cộng tác giữa các bên.

Cuộc họp này được đóng khung trong bốn giờ cho các Sprint có độ dài một tháng. Sprint ngắn hơn thì thời gian họp rút bớt cho phù hợp. Ví dụ, một Sprint hai tuần chỉ cần đến hai giờ cho buổi Sơ kết Sprint.

Buổi Sơ kết Sprint có một số đặc điểm sau:

* Product Owner nhận biết phần nào là hoàn thành và phần nào chưa hoàn thành.
* Team thảo luận những điều thuận lợi trong Sprint vừa qua, những khó khăn mà nhóm đã trải qua, và cách thức giải quyết các vấn đề đó.
* Team trình diễn các phần việc đã hoàn thành và trả lời các câu hỏi mọi người về gói tăng trưởng.
* Product Owner trao đổi về Product Backlog. Dựa trên tiến độ hiện thời, Product Owner đưa ra dự đoán ngày hoàn thành dự án.
* Toàn bộ nhóm thảo luận về những gì sẽ làm, nhờ đó buổi Sơ kết Sprint cung cấp các giá trị đầu vào cho buổi họp kế hoạch Sprint tiếp theo.

Kết quả của cuộc họp Sơ kết Sprint là một bản Product Backlog đã được cập nhật, với các hạng mục dự định sẽ được triển khai trong Sprint tới. Product Backlog có thể được điều chỉnh toàn diện để thích ứng với các cơ hội mới.

**Cải tiền Sprint**

Buổi họp cải tiến Sprint (Sprint Retrospective) là cơ hội để Nhóm Scrum tự thanh tra và đưa ra kế hoạch cho các cải tiến trong Sprint tiếp theo. Buổi họp cải tiến Sprint được tổ chức ngay sau sơ kết Sprint và trước khi cuộc họp kế hoạch Sprint tiếp theo diễn ra. Cuộc họp này được đóng khung trong phạm vi ba giờ cho các Sprint một tháng. Sprint ngắn hơn thì cuộc họp sẽ được rút ngắn lại cho phù hợp.

Mục đích của cuộc họp cải tiến Sprint là để:

* Thanh tra lại tất cả các yếu tố trong Sprint vừa diễn ra, từ con người, các mối quan hệ, quy trình, và công cụ.
* Nhận biết và xếp đặt lại các hạng mục chủ chốt đã được thực hiện tốt, và các cải tiến dự định.
* Tạo ra một kế hoạch để triển khai các cải tiến cách thức làm việc của Nhóm Scrum.

Scrum Master động viên Nhóm Scrum để cải tiến, trong phạm vi khung làm việc Scrum, quy trình phát triển và các biện pháp thực hành để nâng cao hiệu quả cho Sprint tiếp theo. Trong cuộc họp cải tiến Sprint , Nhóm Scrum sẽ lập kế hoạch để gia tăng chất lượng sản phẩm bằng việc định nghĩa lại dịnh nghĩa “hoàn thành” khi cần thiết.

Kết thúc cuộc họp cải tiến Sprint, nhóm Scrum phải nhận biết ra các cải tiến sẽ được triển khai trong Sprint tới.

### Các đồ nghề trong Scrum

**Product Backlog**

Product Backlog là một danh sách sắp thứ tự tất cả những gì cần thiết của sản phẩm. Product Owner là người chịu trách nhiệm về Product Backlog, nội dung của nó, sự hiện diện, và thứ tự các hạng mục trong đó.

Product Backlog có thể không bao giờ hoàn chỉnh. Phiên bản sớm nhất của Product Backlog chỉ cho thấy các yêu cầu được tìm hiểu rõ ràng từ lúc đầu tiên. Product Backlog sẽ tiến hóa cùng với với sản phẩm và môi trường mà nó sẽ được sử dụng. Product Backlog là động, nó thay đổi thường xuyên để nhận biết những gì mà sản phẩm cần phải có để có tính cạnh tranh và hữu ích. Chừng nào sản phẩm còn đó, thì Product Backlog cũng hiện diện.

Product Backlog liệt kê tất cả các chức năng, yêu cầu, cải thiện, vá lỗi cần thiết để làm nên sản phẩm trong tương lai. Các hạng mục trong Product Backlog được mô tả với các thuộc tính như: mô tả, thứ tự, và ước lượng.

Product Backlog thường được sắp xếp theo các giá trị, độ rủi ro, độ ưu tiên, và sự cần thiết. Các hạng mục đứng đầu danh sách sẽ trực tiếp điều khiển các hoạt động phát triển. Càng ở thứ tự cao hơn, các hạng mục càng được quan tâm nhiều hơn, và được tập trung nỗ lực nhiều hơn vì chính giá trị của chúng.

Khi sản phẩm được đưa vào sử dụng và mang lại giá trị, thị trường sẽ cung cấp các phản hồi, Product Backlog sẽ trở thành một danh sách lớn hơn và toàn diện hơn. Nhu cầu thì không ngừng thay đổi, vì thế một Product Backlog luôn luôn thay đổi. Sự thay đổi trong các yêu cầu nghiệp vụ, điều kiện thị trường, hay công nghệ có thể dẫn đến các thay đổi trong Product Backlog.

Trong trường hợp nhiều Nhóm Scrum làm việc với nhau trên cùng một sản phẩm. Một Product Backlog được dùng để mô tả những công việc tới đây của Sản phẩm. Khi đó, các hạng mục có thể được nhóm lại theo một tính chất nào đó.

Việc “làm mịn” Product Backlog là hoạt động thêm vào các chi tiết, ước lượng, và trình tự của các hạng mục trong Product Backlog. Đây là quá trình liên tục, theo đó Product Owner và Team thảo luận về các chi tiết của từng hạng mục. Trong suốt quá trình làm mịn này, các hạng mục liên tục được xem xét và rà soát cẩn thận. Tuy nhiên, chúng có thể được cập nhật tại bất kì thời điểm nào bởi Product Owner.

**Sprint Backlog**

Sprint Backlog là tập hợp các hạng mục Product Backlog được lựa chọn để phát triển trong Sprint, kèm theo một kế hoạch để chuyển giao phần tăng trưởng của sản phẩm và hiện thực hóa mục tiêu Sprint. Sprint Backlog là một bản dự báo của Team về những chức năng sẽ có trong phần tăng trưởng sẽ được chuyển giao.

Sprint Backlog xác định công việc mà Team phải làm để biến các hạng mục Product Backlog thành phần tăng trưởng đạt tiêu chuẩn “hoàn thành”. Sprint Backlog cho thấy tất cả những việc Team cần phải làm để tiến tới mục tiêu Sprint.

Sprint Backlog là một kế hoạch với chi tiết vừa đủ để những thay đổi về tiến độ công việc có thể nhìn thấy được trong các cuộc họp Scrum hằng ngày. Team chỉnh sửa Sprint Backlog trong suốt Sprint, và Sprint Backlog sẽ được cập nhật trong suốt thời gian đó. Sự cập nhật này xảy ra trong khi Team thực hiện theo kế hoạch của, và hiểu rõ hơn về các công việc cần thiết để đạt mục tiêu Sprint.

Mỗi khi có thêm việc mới, Team đưa vào Sprint Backlog. Khi công việc bắt đầu hay kết thúc, giá trị ước lượng về thời gian còn lại để hoàn tất công việc được cập nhật. Khi có phần nào đó của kế hoạch là không cần thiết, chúng sẽ bị bỏ đi. Chỉ có Team mới có thể thay đổi Sprint Backlog trong Sprint. Sprint Backlog là một bức tranh thời gian thực về công việc mà Team lên kế hoạch để hoàn thành trong Sprint, và nó cơ bản thuộc về Team.

**So sánh Product Backlog và Sprint Backlog**

Product backlog và Sprint backlog có nhiều điểm chung nhưng khác mục đích sử dụng. Product backlog được quản lí bởi product owner và cung cấp một cái nhìn ở mức cao về công việc mà team phát triển phải hoàn thành để tạo ra sản phẩm. Product owner đánh giá user story trong Product Backlog và cung cấp các thông tin chi tiết phù hợp trong buổi họp kế hoạch Sprint để Team phát triển có thể đánh giá và hiện thực từng Story.

Ngược lại, Team của bạn tạo Sprint Backlog, chứa danh sách chi tiết tất cả các Task mà Team cần phải hoàn thành để kết thúc các User Story trong Sprint. Trong Product Backlog, Team của bạn đánh giá User Story bằng Story Point còn trong Sprint Backlog, Team đánh giá Task bằng số giờ thực hiện.

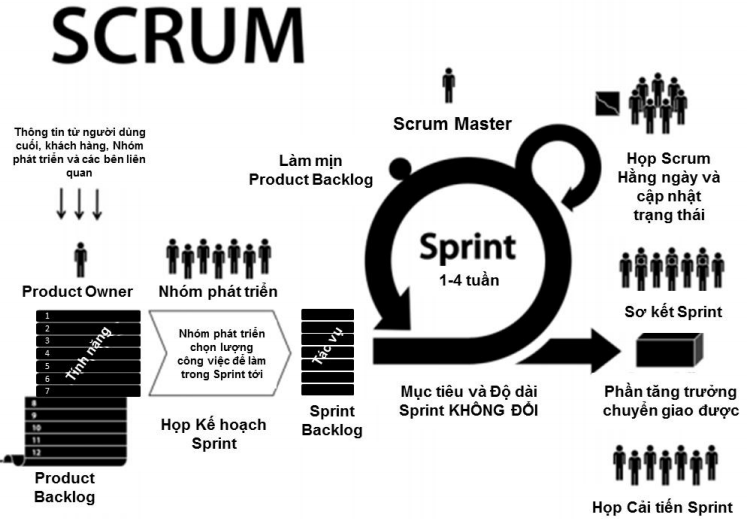
Product Owner cập nhật Product Backlog hành tuần, nhưng Team phải cập nhật Sprint Backlog hàng ngày.

Product Owner duy trình Product Backlog suốt dự án, nhưng Team tạo Sprint Backlog mới vào mỗi Sprint.

Bảng sau tóm tắt những điểm khác nhau giữa Product Backlog và Sprint Backlog.

|  |  |  |
| --- | --- | --- |
| **Mục** | **Product Backlog** | **Sprint Backlog** |
| Mức độ chi tiết | Ít chi tiết hơn | Rất chi tiết |
| Đơn vị ước lượng | Story points | Số giờ |
| Chủ sở hữu | Product owner | Team phát triển |
| Duyệt lại | Hàng tuần | Hàng ngày |
| Kéo dài | Dự án | Sprint |

### Quy trình triển khai Scrum

****

Hình .: Quy trình triển khai Sprint trong Scrum

*(Nguồn: Sách Scrum Căn bản Phiên bản 1.2)*

**Bước 1:** Product Owner đưa ra danh sách các chức năng (User Story) được sắp xếp theo độ quan trọng. Danh sách này tồn tại và cập nhật trong suốt quá trình phát triển sản phẩm

**Bước 2:** Bắt đầu mỗi Sprint, nhóm Scrum sẽ tiến hàng cuộc họp kế hoạch Sprint.

* Team sẽ chọn ra các User Story để thực hiện trong Sprint. Trong suốt quá trình thực hiện Sprint thì các User Story không được thay đổi.
* Phân chia các User Story thành các Task
* Kết quả của bước này là một bản Sprint Backlog gồm danh sách các User Story sẽ được thực hiện trong Sprint cùng các Task của nó.

**Bước 3:** Mỗi ngày Team tổ chức họp hàng ngày để thanh tra quá trình thực hiện, thực hiện các thay đổi nếu cần để hoàn thành công việc. Đồng thời đây cũng là thời điểm để cập nhật các Task và Sprint Backlog

**Bước 4:** Trước khi Sprint kết thúc, nhóm Scrum tiến hành họp sơ kết Sprint với các bên liên quan, trình bày về những việc đã làm, giới thiệu kết quả, xem xết lại những việc chưa hoàn thành.

**Bước 5:** Tiếp theo, nhóm Scrum sẽ tổ chức cuộc họp cải tiến Sprint. Mọi người xem xét lại Sprint vừa diễn ra và nhận phản hồi, ý tưởng để cải tiến trong Sprint tiếp theo.

Sau khi kết thú Sprint thì Product Owner sẽ cập nhật lại sự thay đổi trong Product Backlog. Lúc này đội ngũ phát triển đã sẵn dàng để bắt đầu Sprint tiếp theo. Quy trình này cứ lặp đi lặp lại cho đến khi hoàn thành tất cả các yêu cầu của khách hàng.

### Điểm mạnh của Scrum

Điểm mạnh nhất đó là việc linh hoạt, dự án không được cố định từ đầu về thời gian hoàn thành hay những yêu cầu mà nó sẽ được xác định khi phát triển thực tế.

Phân phối sản phẩm mềm dẻo: nội dung sản phẩm chuyển giao được xác định linh hoạt theo môi trường sử dụng thực tế. Thời gian biểu linh hoạt: có thể muộn hoặc sớm hơn so với kế hoạch ban đầu.

Chất lượng sản phẩm tốt và giảm rủi ro sản xuất, chi phí thấp. Khả năng trao đổi giữa khách hàng và nhà phát triển, giữa những thành viên trong đội được đặt lên mức cao.

Tốc độ phát triển nhanh, tiết kiệm thời gian. Việc chuẩn bị hành động cho những thay đổi trong quá trình phát triển tốt hơn vì hầu như hàng ngày luôn có những buổi họp đánh giá lại ở những vòng lặp phát triển.

Các lỗi và các vấn đề được phát hiện sớm hơn rất nhiều so với các phương pháp truyền thống bởi vì khách hàng được tham gia đánh giá rất nhiều và đầu ra của sản phẩm rất nhanh. Và khi đi sai hướng, có thể hủy ngay Sprint đó để quay lại với bản kế hoạch.

### So sánh với các phương pháp truyền thống

|  |  |  |  |
| --- | --- | --- | --- |
| **Đặc điểm** | **Waterfall** | **Spiral** | **Scrum** |
| Xác định các giai đoạn phát triển | Bắt buộc | Bắt buộc | Chỉ có giai đoạn lập kế hoạch và kết thúc |
| Sản phẩm cuối cùng | Được xác định trong quá trình lập kế hoạch | Được xác định trong quá trình lập kế hoạch | Xác định trong quá trình xây dựng dự án |
| Chi phí sản phẩm | Được xác định trong quá trình lập kế hoạch | Thay đổi cục bộ | Xác định trong quá trình xây dựng dự án |
| Ngày hoàn thành sản phẩm | Được xác định trong quá trình lập kế hoạch | Thay đổi cục bộ | Xác định trong quá trình xây dựng dự án |
| Đáp ứng với môi trường sử dụng | Trong kế hoạch ban đầu | Trong kế hoạch ban đầu | Xuyên suốt từ kế hoạch đến xây dựng và kết thúc |
| Kinh nghiệm trao đổi | Đào tạo trước cho đến khi bắt tay làm dự án | Đào tạo trước cho đến khi bắt tay làm dự án | Thực hiện trong quá trình làm dự án |
| Khả năng thành công | Thấp | Trung bình thấp | Cao |

# XÂY DỰNG CÔNG CỤ

## Phân tích yêu cầu

### Mô tả chi tiết công cụ

Nhóm đặt tên công cụ là Scrumreal nhấn mạnh tới khả năng cập nhật thời gian thực của công cụ.

Scrumreal là một công cụ hỗ trợ phát triển phần mềm theo mô hình Scrum.

Công cụ chia đội ngũ tham gia xây dựng dự án thành 4 vai trò sau: Product Owner, Project Leader, Scrum Master, Team mỗi vai trò có quyền hạn nhất định trong công cụ.

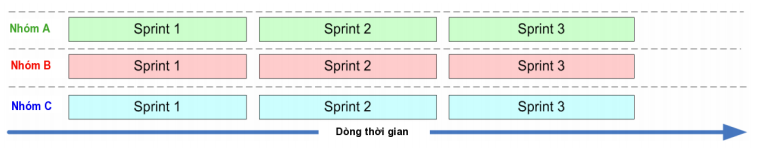
Vai trò Project Leader được thêm vào để quản lí nhân lực khi có nhiều nhóm Scrum cùng tham gia vào dự án, làm việc cùng với Product Owner để xây dựng các user story và cũng là người phân công user story cho từng nhóm. Project Leader phải là người am hiểu rất rõ về Scrum (Scrum Master), họ không thuộc về bất kì một nhóm nào, cũng không phải là lãnh đạo của toàn đội ngũ và trong một project chỉ có một Project Leader duy nhất và một Project Leader có thể làm việc với nhiều dự án khác nhau. Trong đội ngũ phát triển, Project Leader là người duy nhất được quyền phân bổ nhân lực, có quyền thay đổi đội ngũ nhân sự của Team nếu cần thiết.

Ngoài ra Project Leader có quyền khởi tạo dự án mới, và thêm mới thành viên. Mỗi thành viên tham gia vào dự án cần có các thông tin sau: tên, ngày sinh, hình ảnh, vai trò (Project leader, Product Owner, Team Member, Scrum Master).

Công cụ có thể quản lí nhiều dựa án khác nhau cùng một lúc. Khi tạo dự án mới Project Leader phải cung cấp các thông tin sau: tên dự án, mô tả, ProductOwner, Project Leader, ngày bắt đầu, ngày kết thúc dự kiến và ghi chú bổ sung.

Các dự án theo mô hình Scrum bao gồm các User Sotry. Chỉ có 2 vai trò là Product Owner và Project Leader là có quyền thao tác, chỉnh sửa User Story. Một User Story gồm có các mục sau: tên, độ quan trọng, ước tính ban đầu, cách demo, lưu ý, phân loại, thành phần kĩ thuật, người yêu cầu. Ước tính ban đầu có đơn vị điểm là Story Point được tính bằng cách hỏi đội dự án “nếu bạn có thể lấy một số lượng người tùy ý đối với story này (không quá ít và không quá nhiều, thường là 2 người) sau đó vào phòng làm việc một cách hết sức tập trung, sau bao nhiêu ngày các bạn sẽ có thể bước ra khỏi phòng làm việc với một sản phẩm được coi là hoàn thành, có thể demo được, đã được test, có thể phát hành được?”. Nếu câu trả lời là “với 3 người ở trong phòng làm việc tập trung sẽ mất khoảng 4 ngày” thì ước tính cho story đó là 12 điểm. Điều quan trọng không phải để bạn xác định được ước tính chính xác tuyệt đối (ví dụ một story 2 điểm sẽ mất 2 ngày để hoàn thành), điều quan trọng là cần ước tính với độ chính xác tương đối (ví dụ một story 2 điểm sẽ chiếm một nửa khối lượng công việc so với story 4 điểm).

Sau khi có user story, Project Leader sẽ ước lượng quy mô, yêu cầu của dự án, tình hình nhân lực từ đó xây dựng đội ngũ phát triển. Việc xây dựng này chỉ là ban đầu và có thể thực hiện các thay đổi và cập nhật nếu cần thiết. Việc thay đổi và cập nhật nhân sự này chỉ duy nhất Project Leader có quyền. Mỗi nhóm sẽ gồm các mục sau: tên nhóm, mô tả, các thành viên trong nhóm, Scrum Master của nhóm.

Sprint được tiến hành đồng bộ giữa các nhóm giống như hình sau:

Hình .: Tiến hành đồng bộ Sprint giữa các nhóm

*(Nguồn: Scrum và Xp từ những chiến hào)*

Việc tiến hành đồng bộ các Sprint mang lại những lợi ích sau: có thời điểm sắp xếp lại các nhóm (xảy ra giữa các Sprint – tránh việc sắp xếp lại nhân sự nhóm khi đang thực hiện Sprint), các nhóm có thể cùng họp kế hoạch Sprint từ đó sự cộng tác giữa các nhóm tốt hơn, ít công sức quản trị hơn (ít buổi họp kế hoạch, ít buổi demo…). Project Leader là người khởi tạo Sprint, một Sprint khi được tạo ra sẽ cho tất cả các eam trong dự án cùng thực hiện. Mỗi Sprint sẽ gồm: tên, mô tả, ngày bắt đầu, ngày kết thúc.

Từ buổi họp kế hoạch Sprint được tổ chức với sự tham gia của tất cả các thành viên thực hiện dự án (Product Owner, Project Leader, Scrum Master, các Team). Thường thì Team sẽ thảo luận với Product Owner và chọn lựa các story đưa vào Sprint nhưng khi có nhiều team, cần phải có vai trò Project Leader để quản lí tránh việc các nhóm cùng làm 1 Story. Lúc này Project Leader sẽ là người duy nhất có quyền phân công story cho nhóm trong Sprint (sau khi đã họp Sprint và thống nhất ý kiến).

Sau khi các Team cùng Project Leader và Product Owner đã thống nhất trong Sprint hiện tại của nhóm có các story nào rồi, nhóm sẽ thực hiện chia nhỏ Story thành các Task. Việc này hoàn toàn do nhóm thực hiện và không có bất cứ sự can thiệp nào khác. Mỗi task sẽ gồm các mục sau: tên, mô tả, thời gian thực hiện, người thực hiện, trạng thái (cần làm, đang làm, cần test, và hoàn thành), độ quan trọng và phân loại. Các Task này cùng trạng thái của chúng sẽ được dùng để thể hiện trong bảng Task board (dưới dạng bảng Kanban tool). Mỗi task chỉ phân công cho một người duy nhất thực hiện (dễ hơn cho việc quản lí).

Trong trường hợp Sprint trước đó có các task chưa hoàn thành thì khi thì story chứa các task chưa hoàn thành đó sẽ được chuyển qua danh sách user story chưa hoàn thành (nhưng vẫn giữa nguyên tiến độ của các task nó chứa). Project Leader có thể phân công story này vào Sprint kế tiếp.

Công cụ lưu trữ tất cả các thay đổi tới dự án, bất kì một cập nhật nào đều được ghi lại (ví dụ: thêm mới hay sửa User Story, thêm Task hay xóa Task…) vào trong mục Activity nhằm đảm bào trách nhiệm của các thành viên thực hiện thay đổi trên công cụ.

Công cụ cung cấp tính năng bình luận, người dùng có thể bình luận ở chi tiết dự án, Sprint, Task..

### Phân tích chi tiết các yêu cầu

|  |  |  |
| --- | --- | --- |
| STT | Yêu cầu | Biểu mẫu |
| 1 | Quản lý danh sách các dự án | BM1 |
| 2 | Quản lí User Story | BM2 |
| 3 | Quản lí Sprint | BM3 |
| 4 | Quản lí Task | BM4 |
| 5 | Quản lí Team | BM5 |
| 6 | Quản lí User | BM6 |

Bảng .: Danh sách các biểu mẫu

#### Quản lí danh sách dự án

*Mô tả:*

Khi có một dự án mới thì người dùng (Scrum Mater) sẽ thêm mới một dự án. Một dự án mới sẽ có các thông tin như sau:

* Mã dự án: là 1 con số tự tăng.
* Tên dự án: một cái tên ngắn có tính mô tả cho dự án.
* Mô tả: giới thiệu về dự án, dự án làm về vấn đề gì, của công ty nào, mục đích của dự án…
* Product Owner: product owner của dự án.
* Project Leader: project leader của dự án.
* Ngày bắt đầu và dự kiến ngày kết thúc của dự án.
* Trạng thái: hoàn thành, hủy, mới khởi tạo, đang thực hiện, tạm hoãn.
* Ghi chú: thông tin bổ sung về dự án.

*Biểu mẫu:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BM1: |  | |  | | Danh Mục Dự Án | | | | | |
| STT | Mã dự án | Tên dự án | | Mô tả | | Project Leader | Product Owner | Ngày bắt đầu | Dự kiến ngày kết thúc | Ghi chú |
| 1 |  |  | |  | |  |  |  |  |  |
| 2 |  |  | |  | |  |  |  |  |  |

*Quy định:* Chỉ có Project leader mới được quyền thêm, sửa xóa các thông tin liên quan tới dự án.

#### Quản lí User Story

*Mô tả:*

* Tên: một cái tên ngắn, mô tả story. Phải chính xác và rõ ràng để Development Team và Product Owner hiểu đang nói vể vấn đề gì và phân biệt các story với nhau.
* Độ quan trọng: Mức độ quan trọng của story do Product Owner đánh giá (khẩn cấp, cao, trung bình, thấp).
* Ước tính ban đầu (story point): Ước tính ban đầu của Development Team về khối lượng việc làm cần thiết để thực hiện story này so với item khác.
* Cách demo: một bản mô tả chi tiết về cách thức demo tính năng này trong buổi demo của Sprint.
* Lưu ý: bất kì thông tin nào khác như dùng để phân loại, tham chiếu tới nguồn tài nguyên khác…
* Phân loại: tag phân loại story. Theo cách này Product Owner có thể rễ dàng lọc ra các story có cùng loại.

*Biểu mẫu:*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| BM2: | Quản lí User Story | | | | | |
| STT | ID | Tên | Độ quan trọng | Ước tính | Cách demo | Lưu ý |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |

*Quy định :* Product Owner có thể thêm Story vào dự án nhưng các giá trị ước tính thì chỉ do Project Leader hoặc Team thực hiện, Product Owner không có quyền.

#### Quản lí Sprint

*Mô tả:*

* Tên: tên của Sprint.
* Mô tả: thông tin mô tả về Sprint.
* Ngày bắt đầu: ngày Sprint bắt đầu.
* Ngày kết thúc: ngày kết thúc Sprint.

*Biểu mẫu:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BM3: | Quản lí Sprint | | | |
| STT | Tên | Mô tả | Ngày bắt đầu | Ngày kết thúc |
| 1 |  |  |  |  |
| 2 |  |  |  |  |

#### Quản lí Task

*Mô tả:*

* Tên: tên của task.
* Mô tả: mô tả task.
* Trạng thái: trạng thái của task (cần làm, đang làm, cần test, và hoàn thành).
* Thởi gian thực hiện: thời gian cần thiết để hoàn thành task.
* Người thực hiện: những người thực hiện task.
* Thời gian đã dùng: thời gian đã làm trên task.
* Thời gian còn lại: thời gian còn lại để hoàn thành task.

*Biểu mẫu:*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| BM4: | Quản lí Task | | | | | | | |
| STT | ID | Tên | Mô tả | Trạng thái | Thời gian thực hiện | Người thực hiện | Thời gian đã dùng | Thời gian còn lại |
| 1 |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |

#### Quản lí Team

*Mô tả:*

* Tên: tên nhóm.
* Mô tả: thông tin mô tả về nhóm.

*Biểu mẫu:*

|  |  |  |
| --- | --- | --- |
| BM5: | Quản lí Team | |
| STT | Tên | Mô tả |
| 1 |  |  |
| 2 |  |  |

#### Quản lí User

*Mô tả:*

* Tên: tên thành viên.
* Mô tả: tuổi thành viên.
* Ngày sinh: ngày sinh của thành viên.
* Mật khẩu: mật khẩu đăng nhập, người dùng có thể thay đổi.
* Hình đại diện: hình đại diện của người dùng.

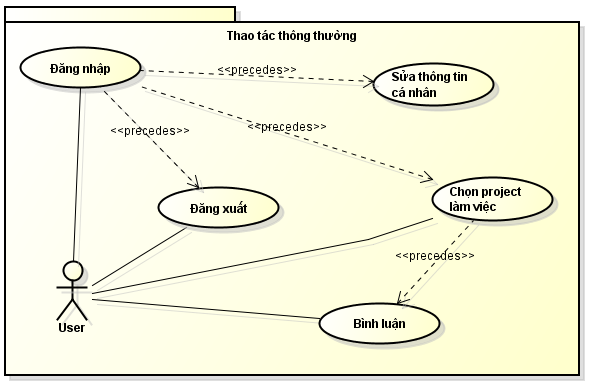
*Biểu mẫu:*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BM3: | Quản lí User | | | | |
| STT | Tên đăng nhập | Tên đầy đủ | Ngày sinh | Mật khẩu | Hình đại diện |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

*Quy định :*

## Sơ đồ UseCase

### Thao tác thông thường



Hình .: Use Case các thao tác thông thường

#### Use Case đăng nhập

* Actor: Người dùng chưa đăng nhập.
* Người dùng truy cập vào hệ thống.
  + Dòng cơ bản: Hệ thống hiển thị màn hình Login, người dùng nhập thông tin đăng nhập (tên đăng nhập, mật khẩu). Sau đó, người dùng nhấp vào nút Login; hệ thống kiểm tra thông tin đăng nhập; nếu thông tin hợp lệ, hệ thống hiển thị trang Project.
  + Dòng thay thế:
    - Người dùng nhập thiếu thông tin: Hệ thống hiển thị thông báo cảnh báo.
    - Thông tin đăng nhập không hợp lệ: Hệ thống thông báo lỗi cho người dùng.

#### Use Case chọn project làm việc

* Tiền điều kiện: người dùng đã đăng nhập thành công vào hệ thống.
* Actor: Người dùng đã đăng nhập.
* Người dùng đăng nhập thành công và hệ thống chuyền người dùng qua trang quản lý Project. Người dùng nhấn vào nút Set current vào project mà họ muốn làm việc trên đó.
  + Dòng cơ bản: hệ thống thiết lập Project được chọn làm project mặc định cho người dùng đó.
  + Dòng thay thế:
    - Có lỗi trong quá trình thiết lập: hệ thống hiển thị thông báo cảnh báo.

#### Use Case đăng xuất

* Actor: Người dùng đã đăng nhập.
* Tiền điều kiện: người dùng đã đăng nhập thành công vào hệ thống.
* Người dùng nhấp vào đường dẫn Log out.
  + Dòng cơ bản: người dùng thoát khỏi hệ thống, hệ thống hiển thị trang Login.

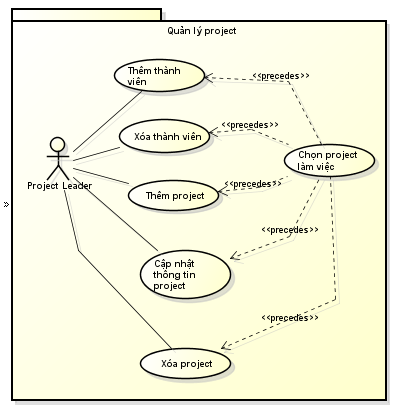
#### Use Case bình luận

* Actor: Người dùng đã đăng nhập.
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Người dùng nhập chọn tab Comment khi xem chi tiết task, user story, sprint, project.
  + Dòng cơ bản: hệ thống hiển thị khung bình luận cho người dùng, người dùng nhập nội dung bình luận rồi nhấn nút Comment, hệ thống cập nhật và hiển thị nội dung bình luận, đồng thời lưu Activity bình luận.
  + Dòng thay thế:
    - Nếu người dùng không nhập nội dung mà nhấn bình luận: hệ thống thông báo lỗi cho người dùng.

#### Use Case sửa thông tin cá nhân

* Actor: Người dùng đã đăng nhập.
* Tiền điều kiện: người dùng đã đăng nhập thành công vào hệ thống.
* Người dùng chọn menu sửa thông tin cá nhân.
  + Dòng cơ bản: hệ thống hiển thị trang Profile. Người dùng tiến hành thay đổi thông tin cá nhân rồi nhấn nút Save. Hệ thống kiểm tra thông tin nhập liệu của người dùng; nếu thông tin hợp lệ hệ thống lưu thông tin vào cở sở dữ liệu.
  + Dòng thay thế:
    - Nếu thông tin nhập liệu của người dùng không hợp lệ: hệ thống thông báo lỗi.

### Quản lí project



Hình .: Use Case quản lí Project

#### Use Case thêm thành viên

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader nhấn vào đường dẫn People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực. Project Leader nhấn Thêm thành viên.
* Dòng cơ bản: hệ thống hiển thị khung thêm thành viên. Project Leader nhập các thông tin của thành viên (tên, tuổi, email….) rồi nhấn Thêm thành viên. Hệ thống kiểm tra thông tin nhập liệu, nếu thông tin phù hợp, hệ thống tạo thành viên mới.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

#### Use Case xóa thành viên

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader vào trang quản lí nhân lực cho project. Project Leader nhấn vào tên User sau đó nhấn vào Delete.
* Dòng cơ bản: hệ thống thực hiện xóa thành viên.
* Dòng thay thế:
  + Nếu Project Leader nhấn Hủy: hệ thống không thực hiện cập nhật, hiển thị trang hiện tại.

#### Use Case thêm project

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader nhấn vào menu Project. Hệ thống hiển thị trang quản lí dự án. Project Leader nhấn Add Project trên trang quản lí dự án.
* Dòng cơ bản: hệ thống hiển thị khung thêm dự án mới. Project Leader nhập thông tin cho dự án rồi nhấn Add project trong khung thêm Project. Hệ thống kiểm tra thông tin nhập liệu; nếu thông tin hợp lệ, hệ thống tạo dự án mới.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

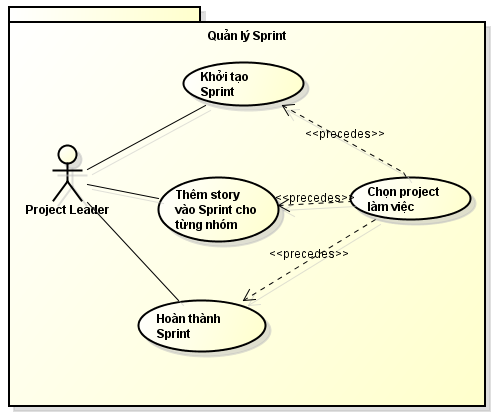
#### Use Case cập nhật thông tin project

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader vào trang quản lí dự án, hệ thống hiển thị danh sách dự án. Project Leader chọn sửa một dự án.
* Dòng cơ bản: hệ thống hiển thị khung sửa dự án. Project Leader tiến hành thay đổi thông tin của dự án rồi nhấn nút Save project information. Hệ thống tiến hành kiểm tra thông tin nhập liệu; nếu thông tin hợp lệ, hệ thống cập nhật dự án.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: Hệ thống thông báo lỗi.

#### Use Case cập nhật thông tin project

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader vào trang quản lí dự án, hệ thống hiển thị danh sách dự án. Project Leader chọn sửa một dự án. Hệ thống hiển thị khung sửa dự án. Project Leader nhấp vào Delete.
* Dòng cơ bản: hệ thống hiển thị thông báo xác nhận. Nếu người dùng nhấn Yes, hệ thống cập nhật trạng thái và ngày kết thúc của dự án.
* Dòng thay thế:
  + Nếu người dùng nhấn No: hệ thống không thực hiện việc cập nhật thông tin project.

### Quản lí Sprint



Hình .: Use Case quản lí Sprint

#### Use Case khởi tạo Sprint

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu Sprint trên thanh menu chính. Hệ thống hiển thị trang quản lí Sprint cho dự án. Project Leader nhấn Add sprint.
* Dòng cơ bản: hệ thống hiển thị khung tạo Sprint. Project Leader tiến hành nhập liệu các thông cần thiết của Sprint sau đó nhấn Add sprint. Hệ thống kiểm tra thông tin nhập liệu; nếu thông tin hợp lệ, hệ thống tạo Sprint mới.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

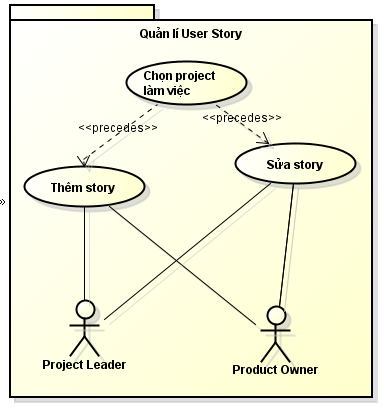
#### Use Case thêm Story vào Sprint

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu Sprint trên thanh menu chính. Hệ thống hiển thị trang quản lí Sprint cho dự án. Project Leader chọn Story muốn đưa vào Sprint, sau đó kéo thả story vào danh sách story của nhóm sẽ thực hiện
* Dòng cơ bản: hệ thống cập nhật danh sách story cho nhóm.
* Dòng thay thế:
  + Nếu tổng số story point của các story được phân công cho Team đã vượt quá tốc độ ước tính của Team, hệ thống sẽ xuất hiện cảnh báo.

#### Use Case hoàn thành Sprint

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu Sprint trên thanh menu chính. Hệ thống hiển thị trang quản lí Sprint cho dự án. Project Leader nhấp vào nút Complete Sprint.Hệ thống hiển thị thông báo xác nhận. Nếu trong Sprint còn tồn tại các Story chưa hoàn thành, hệ thống hiển thị thông báo danh sách tên các Story đó cùng với thống báo xác nhận.
* Dòng cơ bản: nếu người dùng đồng ý, hệ thống cập nhật trạng thái của Sprint thành hoàn thành và những Story chưa hoàn thành sẽ được chuyển lại vào danh sách Story để thêm vào các Sprint khác.
* Dòng thay thế: nếu người dùng không đồng ý thay đổi, hệ thống không thực hiện cập nhật trên dữ liệu và hiển thị trang Sprint.

### Quản lí User Story



Hình .: Use Case quản lí User Story

#### Thêm Story

* Actor: Product Owner, Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Product Owner vào trang User story, sau đó chọn Add story.
* Dòng cơ bản: hệ thống hiển thị khung thêm user story. Actor nhập thông tin của story; hệ thống kiểm tra thông tin; nếu thông tin hợp lệ hệ thống lưu story.
* Dòng thay thế:
  + Thông tin nhập liệu cho user story không hợp lệ: hệ thống thông báo lỗi.

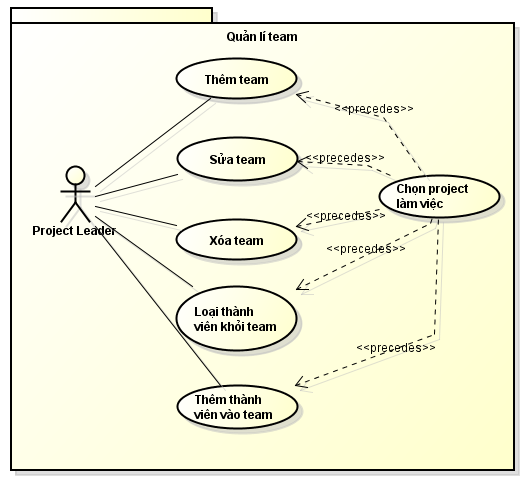
#### Sửa Story

* Actor: Product Owner, Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Actor vào trang User Story chọn sửa một Story
* Dòng cơ bản: hệ thống hiển thị khung sửa story. Actor tiến hành nhập liệu, sau đó nhấn Save story. Hệ thống kiểm tra dữ liệu nhập vào; nếu dữ liệu hợp lệ hệ thống cập nhật story.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

#### Xóa Story

* Actor: Product Owner, Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Actor vào trang User Story chọn sửa một Story. Hệ thống hiển thị khung chi tiết Story, actor nhấn Delete, hệ thống xuất hiện thông báo xác nhận.
* Dòng cơ bản: actor xác nhận, hệ thống tiến hành xóa Story khỏi dự án.
* Dòng thay thế:
  + Actor hủy xác nhận: hệ thống không thực hiện cập nhật.

### Quản lí Team



Hình .: Use Case quản lí Team

#### Thêm Team

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực cho dự án. Project Leader nhấn Add team.
* Dòng cơ bản: hệ thống hiển thị khung tạo nhóm, Project Leader nhập thông tin nhóm sau đó nhấn Add team. Hệ thống kiểm tra thông tin nhập liệu; nếu dữ liệu hợp lệ, hệ thống tạo Team mới.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

#### Sửa Team

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực cho dự án. Project chọn xem chi tiết Tea Team bằng cách nhấp vào tên của Team. Hệ thống hiển thị khung Edit Team; Project Leader sửa đổi thông tin và nhấn Save team.
* Dòng cơ bản: hệ thống cập nhật thông tin về Team.
* Dòng thay thế:
  + Nếu Project Leader nhập Cancel: hệ thống không thực hiện thay đổi gì và hiển thị trang hiện tại.

#### Xóa Team

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực cho dự án. Project Leader chọn xem chi tiết Team bằng cách nhấp vào tên của Team. Hệ thống hiển thị khung Edit Team, Project Leader nhấn Delete.
* Dòng cơ bản: Hệ thống hiển thị thông báo xác nhận; nếu Project Leader chọn Đồng ý, hệ thống xóa nhóm khỏi dự án.
* Dòng thay thế:
  + Nếu Project Leader chọn Cancel: hệ thống không thực hiện thay đổi gì và hiển thị trang hiện tại.
  + Nếu nhóm đang thực hiện dự án: hệ thống sẽ yêu cầu Project Leader bỏ các task đã phân công cho nhóm ra mới có thể hủy nhóm.

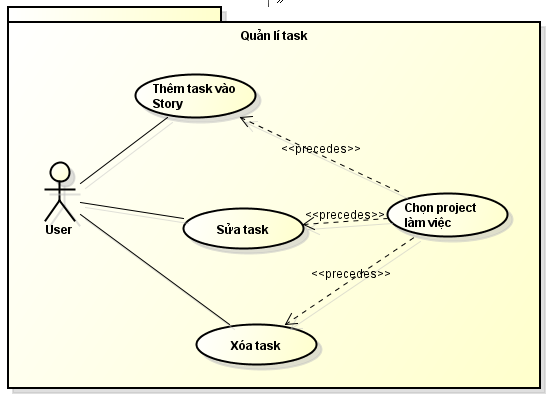
#### Loại thành viên khỏi Team

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực cho dự án. Project Leader chọn thành viên muốn xóa khỏi nhóm và kéo thả khỏi danh sách nhân lực trong Team.
* Dòng cơ bản: Hệ thống xóa thành viên khỏi Team.
* Dòng thay thế:
  + Hệ thống gặp lỗi khi thực hiện: xuất hiện thông báo cho người dùng.

#### Thêm thành viên vào Team

* Actor: Project Leader
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Project Leader chọn menu People trên thanh menu chính. Hệ thống hiển thị trang quản lí nhân lực cho dự án. Project Leader kéo thả thành viên dự án từ khung danh sách nhân viên vào Team mong muốn.
* Dòng cơ bản: Hệ thống cập nhật danh sách nhóm.
* Dòng thay thế:
  + Hệ thống gặp lỗi khi thực hiện: xuất hiện thông báo cho người dùng.

### Quản lí Task



Hình .: Use Case quản lí Task

#### Thêm Task vào User Story

* Actor: Scrum Master
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Actor xem chi tiết một Story, sau đó chọn tab Tasks rồi nhấn Add task.
* Dòng cơ bản: hệ thống hiển thị khung thêm task cho User Story được chọn. Scrum Master nhập dữ liệu và nhấn Add task; hệ thống kiểm tra thông tin nhập liệu; nếu thông tin hợp lệ, hệ thống cập nhật danh sách task của Story.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

#### Sửa Task

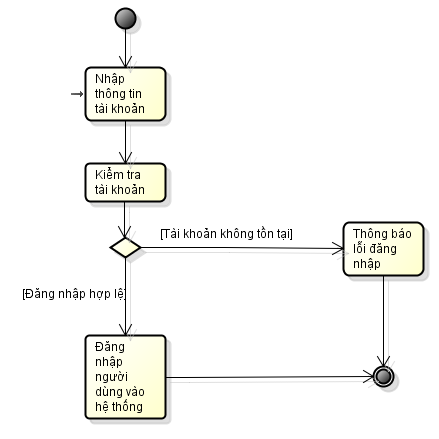
* Actor: Scrum Master
* Actor chọn xem chi tiết task (có thể từ danh sách task của User Story hoặc từ trang Taskboard).
* Dòng cơ bản: hệ thống hiển thị khung chi tiết task. Scrum Master nhập dữ liệu và nhấn Save task; hệ thống kiểm tra thông tin nhập liệu; nếu thông tin hợp lệ hệ thống cập nhật danh sách task của story.
* Dòng thay thế:
  + Thông tin nhập liệu không hợp lệ: hệ thống thông báo lỗi.

#### Xóa Task

* Actor: Scrum Master, Development Team
* Tiền điều kiện: actor đã thiết lập project làm việc.
* Scrum Master chọn xem chi tiết một task rồi nhấn vào nút Delete.
* Dòng cơ bản: hệ thống hiển thị thông báo xác nhận; nếu Project Leader đồng ý hệ thống thực hiện xóa task.
* Dòng thay thế:
  + Nếu Project Leader hủy xác nhận: hệ thống không thực hiện cập nhật.

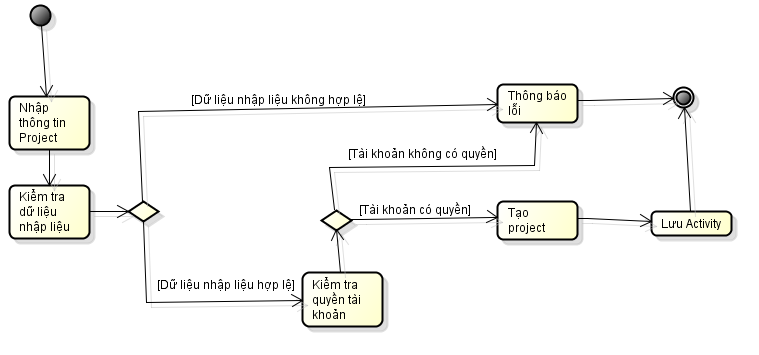
## Sơ đồ Activity

### Chức năng đăng nhập



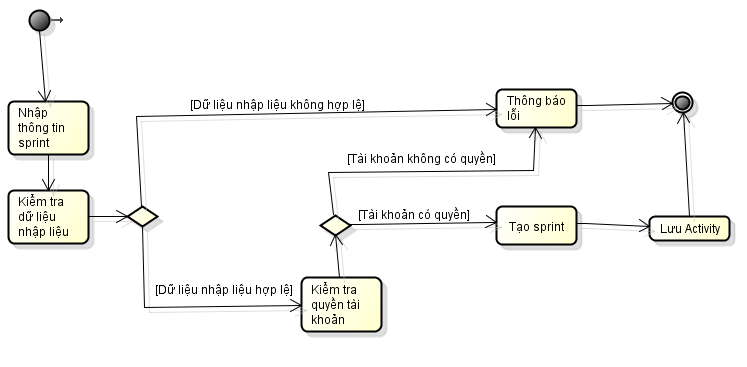
Hình .: Activity diagram đăng nhập

### Chức năng tạo project



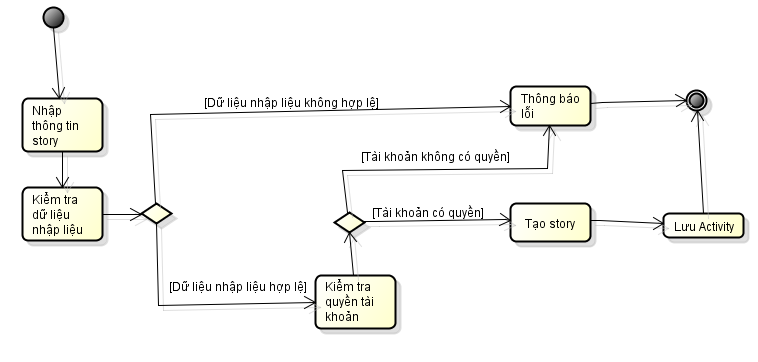
Hình .: Activity diagram tạo project

### Chức năng tạo sprint



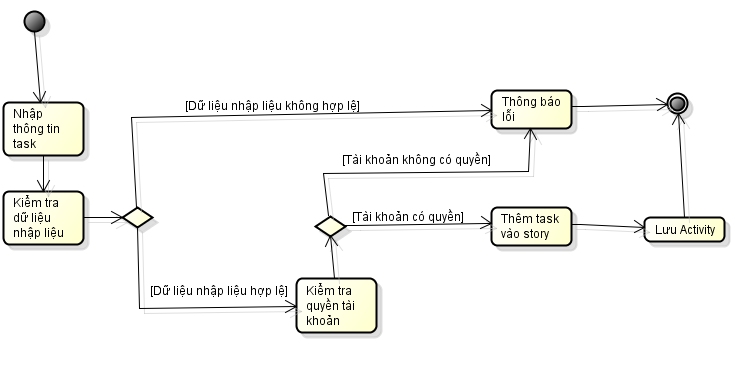
Hình .: Activity diagram tạo sprint

### Chức năng tạo story



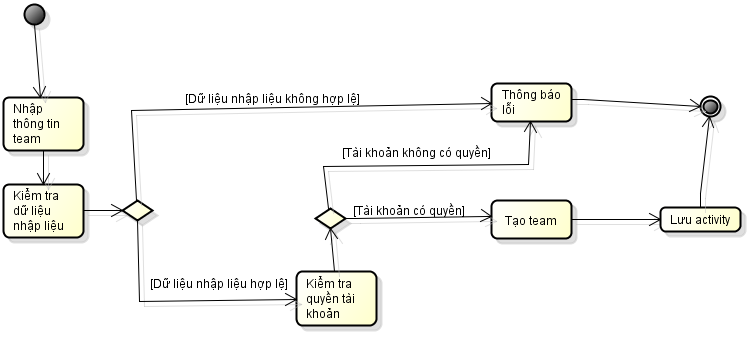
Hình .: Activity diagram tạo story

### Chức năng tạo task



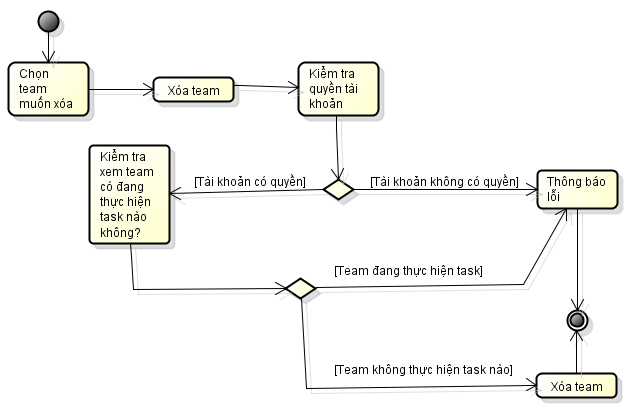
Hình .: Activity diagram tạo task

### Chức năng tạo team



Hình .: Activity diagram tạo team

### Chức năng xóa team



Hình .: Activity diagram xóa team

## Tổ chức cơ sở dữ liệu

### Bảng access\_link

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên thuộc tính** | **Loại** | **Độ dài** | **Ghi chú** | **Ý nghĩa** |
| lid | int | 11 | Not null | Mã đường dẫn |
| path | varchar | 255 | Not null | Đường dẫn |
| type | tinyint | 4 | Not null | Loại đường dẫn (0: thông thường, 1: regular expression) |
| peid | int | 11 | Not null | Mã quyền |

### Bảng activity

### Bảng attach

### Bảng comment

### Bảng permission

### Bảng project

### Bảng project\_user

### Bảng role

### Bảng role\_permission

### Bảng sprint

### Bảng sprint\_team

### Bảng story

### Bảng story\_team

### Bảng task

### Bảng team

### Bảng user

## Thiết kế giao diện

### Chủ đề cấp độ 3

#### Chủ đề cấp độ 4

## Chủ đề cấp độ 2

## Thiết kế dữ liệu

## Thiết kế giao diện

### Chủ đề cấp độ 3

#### Chủ đề cấp độ 4

## Chủ đề cấp độ 2

# TỔNG KẾT

## Kết quả đạt được

## Hướng phát triển

### Chủ đề cấp độ 3

#### Chủ đề cấp độ 4

## Chủ đề cấp độ 2

TÀI LIỆU THAM KHẢO

Theo chuẩn IEEE