

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP. HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN**  
**THỰC TẬP CHUYÊN MÔN**

**ĐỀ TÀI: NGHIÊN CỨU PHƯƠNG PHÁP LUẬN AGILE – SCRUM**

**Sinh viên thực hiện:**

**Họ và tên: Phạm Quỳnh Hương    CQ.61.CNTT    Bộ môn Công nghệ thông tin**

**Giảng viên hướng dẫn: ThS. Trần Phong Nhã & Ks. Nguyễn Thiện Dương**

**TP.Hồ Chí Minh, ngày 01 tháng 07 năm 2023**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**  
**PHÂN HIỆU TẠI TP.HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN**  
**THỰC TẬP CHUYÊN MÔN**  
**ĐỀ TÀI: NGHIÊN CỨU PHƯƠNG PHÁP LUẬN AGILE – SCRUM**

**Sinh viên thực hiện:**

Họ tên : Phạm Quỳnh Hương

Mã sinh viên : 6151071060

Lớp : CQ.CNTT.K61

Ngành đào tạo : Công Nghệ Thông Tin

Email : [6151071060@gmail.com](mailto:6151071060@gmail.com)

Năm thứ: 3/4

**Giảng viên hướng dẫn: ThS. Trần Phong Nhã & Ks. Nguyễn Thiện Dương**

**TP.Hồ Chí Minh, ngày 01 tháng 07 năm 2023**

# LỜI MỞ ĐẦU

Trong thời đại công nghệ và sự thay đổi liên tục, việc áp dụng các phương pháp quản lý dự án linh hoạt và hiệu quả trở thành một yếu tố quan trọng đối với các tổ chức và doanh nghiệp. Trong lĩnh vực phát triển phần mềm, phương pháp Agile và quy trình Scrum đã trở thành một trong những công cụ được sử dụng phổ biến nhất để tăng cường khả năng phản ứng, linh hoạt và tăng cường hiệu suất của các nhóm phát triển.

Phương pháp Agile là một triết lý quản lý dự án đặt trọng tâm vào sự cộng tác, tư duy linh hoạt và sự thích ứng với sự thay đổi. Nó giúp các nhóm phát triển tạo ra sản phẩm có giá trị cao và đáp ứng nhanh chóng đối với yêu cầu thay đổi từ phía khách hàng.

Scrum, một trong những khung làm việc phổ biến trong phương pháp Agile, cung cấp một cách thức cụ thể để tổ chức và quản lý quy trình phát triển phần mềm. Scrum tập trung vào việc chia dự án thành các đợt ngắn gọi là "Sprint" và sử dụng các cuộc họp ngắn gọi là "Daily Scrum" để theo dõi tiến độ và điều chỉnh công việc. Nó cung cấp cơ chế linh hoạt để thích ứng với các thay đổi trong quá trình phát triển và tạo sự minh bạch và sự tương tác tốt giữa các thành viên trong nhóm.

Trên cơ sở đó, nghiên cứu về phương pháp Agile và quy trình Scrum trở nên quan trọng để hiểu rõ các nguyên tắc cơ bản, cách thức triển khai và lợi ích của việc sử dụng chúng trong môi trường phát triển phần mềm. Nghiên cứu này sẽ tập trung vào phân tích chi tiết về phương pháp Agile và quy trình Scrum, từ cách thức tổ chức dự án, vai trò và trách nhiệm của các thành viên trong nhóm, đến quy trình lập kế hoạch, triển khai và đánh giá Sprint. Bằng cách nghiên cứu sâu và hiểu rõ về phương pháp Agile và quy trình Scrum, chúng ta có thể áp dụng chúng một cách hiệu quả và tận dụng tối đa các lợi ích mà chúng mang lại. Do đó, em quyết định thực hiện nghiên cứu đề tài *Nghiên cứu phương pháp Agile quy trình Scrum*— một phương pháp hứa hẹn trong việc nâng cao hiệu suất và linh hoạt trong quy trình phát triển phần mềm.

Báo cáo này được em viết nhằm tổng kết, đúc rút cũng như thực hành việc nghiên cứu, làm đồ án thực tập chuyên môn dựa trên lý thuyết được truyền giảng trên lớp và quá trình học hỏi, tìm tòi bản chuyên sâu của em ngoài thực tế. Qua đây, em xin cảm ơn chân thành đến Thầy Trần Phong Nhã, Cô Trần Thị Dung, Thầy Nguyễn thiện Dương, Thầy Trần Quốc Khánh và các giảng viên bộ môn công nghệ thông tin đã tận tình dạy bảo, giải đáp những thắc mắc của em trong quá trình tìm hiểu và nghiên cứu các vấn đề.

Nội dung báo cáo ngoài việc tham khảo tài liệu, nội dung hoàn toàn mang quan điểm của cá nhân, viết theo văn phong cá nhân. Do đó sẽ không thể tránh khỏi những sai sót không muốn. Rất mong quý thầy cô nghiên cứu tích cực nhận xét để cải thiện.

# MỤC LỤC

<b>CHƯƠNG I : TỔNG QUAN ĐỀ TÀI.....</b>	<b>8</b>
1.1 Tình hình nghiên cứu:.....	8
1.2 Lý do chọn đề tài: .....	8
1.3 Mục tiêu, nội dung, phương pháp nghiên cứu đề tài:.....	8
1.3.1 Mục tiêu nghiên cứu: .....	8
1.3.2 Nội dung nghiên cứu: .....	9
1.3.3 Phương pháp nghiên cứu: .....	9
1.3 Đối tượng và phạm vi nghiên cứu: .....	9
<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....</b>	<b>11</b>
2.1. Tổng quan Agile: .....	11
2.1.1 Đặc điểm của phương pháp luận Agile: .....	13
2.2 Các phương pháp của Agile: .....	16
2.2.1 Scrum: .....	16
2.2.2 Kanban: .....	21
2.2.3 Scrumban: .....	25
2.2.4 Feature-Driven Development (FDD): .....	26
2.2.5 Crystal: .....	27
2.2.6 Extreme Programming: .....	28
2.2.4 So sánh hai mô hình Agile và Waterfall: .....	34
<b>CHƯƠNG 3: KẾT LUẬN VÀ KIẾN NGHỊ .....</b>	<b>i</b>
3.2 Tồn tại: .....	i
3.3 Kiến nghị: .....	i

# CHƯƠNG I : TỔNG QUAN ĐỀ TÀI

## 1.1 Tình hình nghiên cứu:

Tình hình nghiên cứu về mô hình Agile và các phương pháp liên quan đang rất phát triển và được quan tâm rộng rãi trong lĩnh vực quản lý dự án và phát triển phần mềm. Mô hình Agile là một phương pháp quản lý dự án linh hoạt, tập trung vào việc tạo ra giá trị cho khách hàng thông qua việc phân chia dự án thành các giai đoạn ngắn gọi là "sprints" và sử dụng kiểm định và phản hồi liên tục từ khách hàng và các thành viên trong nhóm phát triển.

Các phương pháp Agile phổ biến bao gồm Scrum, Kanban, Extreme Programming (XP), Crystal và nhiều hơn nữa. Các phương pháp này đều nhấn mạnh tính linh hoạt, sự tương tác và sự hợp tác giữa các thành viên trong nhóm phát triển.

Nghiên cứu về mô hình Agile và các phương pháp liên quan rất phong phú và đa dạng. Hiện có nhiều nghiên cứu trên cả mô phỏng và ứng dụng thực tế của Agile trong các lĩnh vực khác nhau.

## 1.2 Lý do chọn đề tài:

- Với nhu cầu hiện nay về việc tìm kiếm các tư liệu cần thiết, phương pháp phát triển phù hợp với điều kiện cũng như chiến lược của công ty, doanh nghiệp. Thì việc tổng hợp được các tư liệu cần thiết về phương pháp và so sánh, kết hợp để tìm ra hướng đi phù hợp cho công ty, doanh nghiệp.

## 1.3 Mục tiêu, nội dung, phương pháp nghiên cứu đề tài:

### 1.3.1 Mục tiêu nghiên cứu:

Mục tiêu của đề tài nghiên cứu mô hình Agile – quy trình Scrum nhằm tìm ra và cung cấp tư liệu phù hợp cho việc tìm kiếm hướng đi hay cách áp dụng hợp lý cho dự án phát triển phần mềm. Với mong muốn tạo ra một mô hình sử dụng ít tài nguyên, chi phí vận hành và hiệu quả cao cho bất kỳ công ty hay doanh nghiệp nào, góp phần vào sự phát triển và thúc đẩy kinh tế thị trường.



### **1.3.2 Nội dung nghiên cứu:**

Nội dung nghiên cứu của đề tài nghiên cứu mô hình Agile và các phương pháp phát triển phần mềm:

Thu thập dữ liệu, biểu đồ: Nghiên cứu này thu thập và phân tích dữ liệu từ khảo sát của các công ty, doanh nghiệp. Qua đó, xây dựng tài liệu tổng hợp chi tiết về mức độ, nhu cầu và các dự án phát triển phần mềm có áp dụng mô hình .

Phân tích các phương pháp: Phân tích kỹ càng các ưu nhược điểm của các phương pháp phổ biến, nghiên cứu sẽ tập trung vào việc theo dõi và phân tích các dự án thành công và thất bại khi áp dụng mô hình.

Đánh giá hiệu suất mô hình: Nghiên cứu cũng sẽ tiến hành đánh giá hiệu suất của mô hình nghiên cứu, đảm bảo rằng các dự án sử dụng các phương pháp chính xác và phù hợp với người dùng. Phân tích sự tương quan giữa phương pháp và nhu cầu điều kiện thực tế của các công ty doanh nghiệp sẽ được thực hiện để đánh giá độ chính xác và hiệu quả của mô hình.

Cải thiện và tối ưu hóa mô hình: Dựa trên kết quả đánh giá, nghiên cứu sẽ tiến hành các bước cải thiện và tối ưu hóa mô hình. Việc này bao gồm việc điều chỉnh tham số, mở rộng phạm vi dữ liệu và nâng cao độ tin cậy và chính xác của mô hình.

### **1.3.3 Phương pháp nghiên cứu:**

Trong đề tài này, em sử dụng kết hợp một số phương pháp nghiên cứu:

Phương pháp định lượng, phương pháp thực nghiệm khảo sát, phương pháp phân tích và tổng hợp.

## **1.3 Đối tượng và phạm vi nghiên cứu:**

Trong đề tài, nghiên cứu về cách hoạt động, áp dụng và các ưu nhược điểm của mô hình Agile và các phương pháp, chủ yếu là Scrum. Ngoài ra, trong đề tài này còn nghiên cứu so sánh các phương pháp của Agile nhằm tìm ra lối đi phù hợp nhất. Việc nghiên cứu mô hình Agile để giải quyết bài toán nghiên cứu phương pháp phù hợp với dự án công ty doanh nghiệp đang cần cũng là vấn đề chú trọng của em trong đề tài này.

Về phạm vi nghiên cứu, xét về mặt không gian em nghiên cứu về các mẫu dữ liệu đã có trên goggle hay các bài báo liên quan đến chủ đề này. Xét về mặt thời gian, nghiên

cứu thực hiện nghiên cứu về đề tài trong giới hạn gần 1 tháng theo thời gian của Trường bắt đầu từ tháng 5 năm 2023 đến tháng 6 năm 2023.

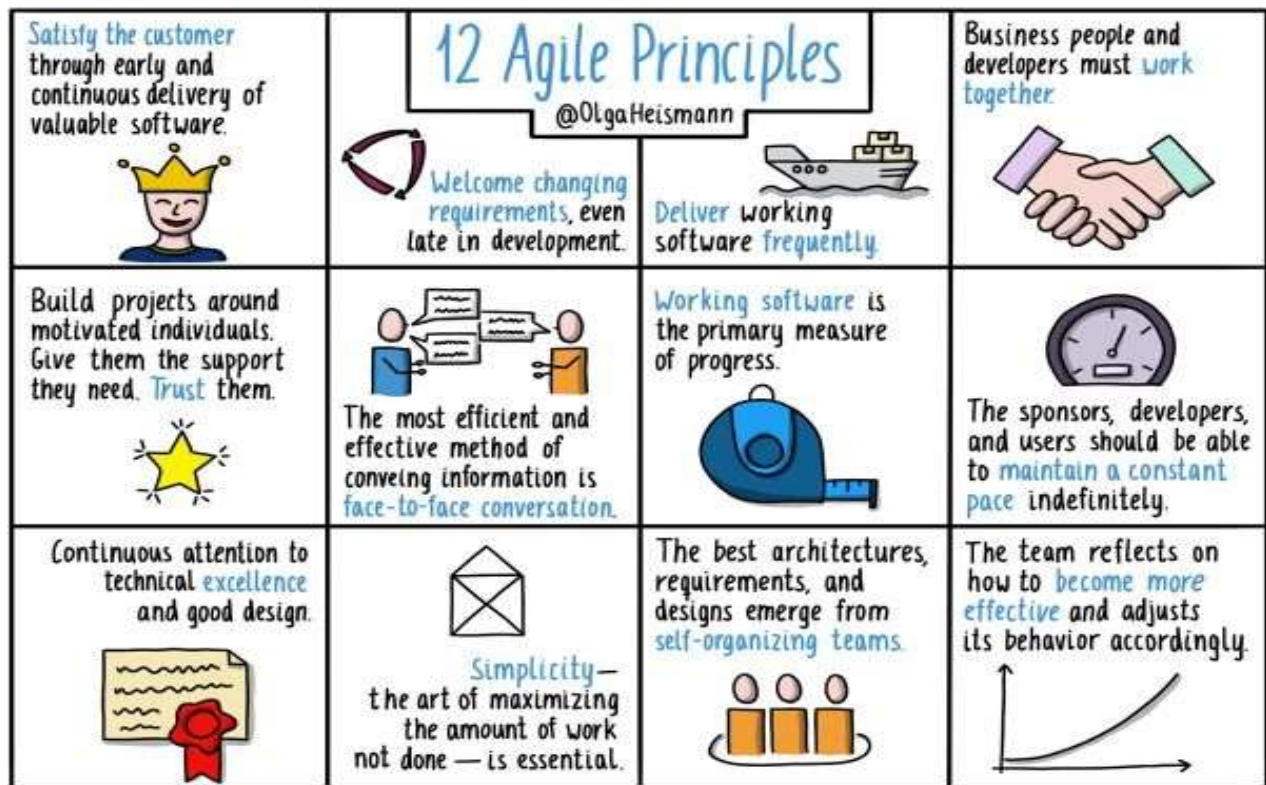
Báo cáo được tổ chức như sau: Chương 1 Tổng quan, Chương 2 Cơ sở lý thuyết, Chương 3 Kết luận và Kiến nghị.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan Agile:

- Gọi là tuyên ngôn Agile Manifesto (tuyên ngôn phát triển phần mềm linh hoạt) - Cuộc khủng hoảng phương pháp luận phát triển phần mềm vào thập kỉ 90 của thế kỉ XX có tỷ lệ thất bại của các dự án phần mềm rất cao. Kết quả là từ ngày 11-13 tháng 2 năm 2001, 17 nhà phát minh và nhà thực hành đã họp với nhau tại bang Utah, Hoa Kỳ để thảo luận về hướng đi mới trong phương pháp luận phát triển phần mềm. Họ đã đi đến thống nhất và cho ra đời bản Tuyên ngôn **Agile** (The Manifesto for **Agile** Software Development) và đánh dấu một xu thế mới trong phát triển phần mềm.
- Ra đời với 4 giá trị cốt lõi: cá nhân và tương tác hơn là quy trình và công cụ; sản phẩm dùng tốt hơn là tài liệu hóa đầy đủ; Cộng tác với khách hàng hơn là đàm phán hợp đồng; Phản hồi với thay đổi hơn là tuân thủ cứng nhắc kế hoạch ban đầu.
- **Agile** thực chất là một triết lý hay một khung tư duy để nhanh chóng thích ứng và phản hồi với thay đổi, từ đó đạt được thành công trong một môi trường liên tục biến động và không chắc chắn. Ra đời trong bối cảnh ngành phát triển phần mềm gặp nhiều thử thách với cách thức phát triển truyền thống theo mô hình Waterfall hoặc theo kế hoạch (plan-driven). Đặc trưng của những phương pháp này là tiếp cận tuyến tính, thực hiện tuần tự các bước theo kế hoạch.
- Các nhà phát triển nhấn mạnh mười hai nguyên lý phía sau Tuyên ngôn **Agile** để giúp các nhà phát triển có được gợi ý trong thực hành và vận dụng các phương pháp **Agile** trong thực tiễn. Các nguyên lý được liệt kê sau đây:
  - Ưu tiên cao nhất 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. Các quy trình linh hoạt tận dụng sự thay đổi trong các lợi thế cạnh tranh của khách hàng.

- Thường xuyên chuyển giao phần mềm chạy tốt tới khách hàng, từ vài tuần đến vài tháng, ưu tiên cho các khoảng thời gian ngắn hơn (trong tầm 1-4 tuần).
- 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 có động lực. Cung cấp cho thành viên môi trường và sự hỗ trợ cần thiết, và tin tưởng để hoàn thành công việc.
- Phương pháp hiệu quả nhất để truyền đạt thông tin tới nhóm phát triển trong nội bộ nhóm phát triển là giao tiếp trực tiếp.
- Phần mềm chạy tốt là thước đo chính của tiến độ.
- Các quy trình linh hoạt thúc đẩy 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 kỹ thuật 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.
- Nhóm phát triển 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.



Hình 2.1. Tổng hợp 12 tuyên ngôn của Agile

### 2.1.1 Đặc điểm của phương pháp luận Agile:

- **Tính lặp (Iterative):** Trong khi dự án thực hiện, các phân đoạn sẽ được lặp đi lặp lại (Iteration hoặc Sprint). Các phân đoạn này diễn ra trong thời gian ngắn (thường 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 và kiểm thử để có được phần nhỏ của sản phẩm. Các phương pháp **Agile** sẽ không lập kế hoạch dài hạn, thay vào đó sẽ phân chia thành những quá trình lập kế hoạch nhỏ, đơn giản và gọn nhẹ.
- **Tính tăng trưởng (Incremental):** Cuối mỗi phân đoạn (Sprint), nhóm phát triển thường cho ra các phần nhỏ của sản phẩm cuối cùng. Các phần nhỏ này thường đáp ứng được các yêu cầu, có khả năng chạy tốt do đã được kiểm thử cẩn thận và có thể sử dụng được ngay. Theo thời gian, các phân đoạn sẽ tiếp nối nhau và tích lũy dần tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn. Khác với mô hình **Waterfall** – vốn chỉ cho phép nhìn thấy sản phẩm tới khi gần hoàn thành dự án, sản phẩm trong dự án **Agile** sẽ được phát triển lớn dần theo thời gian, tăng trưởng cho tới khi đạt được trạng thái đủ để phát hành.

- **Vòng phản hồi ngắn và thích ứng thường xuyên:** Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn, việc lập kế hoạch hay có những điều chỉnh, thay đổi trong quá trình phát triển đều có thể đáp ứng nhanh để phù hợp. Ngoài ra, việc khách hàng được tham gia vào các quy trình phát triển cũng sẽ giúp ích cho việc đáp ứng và thay đổi ngay những yêu cầu khác từ phía khách hàng.
- **Giao tiếp thường xuyên và hiệu quả:** Trong các nhóm Agile luôn đề cao việc giao tiếp thường xuyên và trực diện hơn là việc trao đổi qua tài liệu, giấy tờ. Các nhóm phát triển cũng thường chỉ ở quy mô nhỏ (đối với Scrum là từ 3-9 người), từ đó sẽ đơn giản hóa được quá trình giao tiếp và thúc đẩy hợp tác hiệu quả hơn.
- **Hướng chất lượng:** Đảm bảo chất lượng tuyệt đỉnh luôn là một yêu cầu quan trọng trong triết lý **Agile**. Rất nhiều kỹ thuật và công cụ được sử dụng để hướng đến việc nâng cao chất lượng sản phẩm, chẳng hạn như: Tích hợp Liên tục, Kiểm thử Đơn vị Tự động, Lập trình cặp, Phát triển Hướng Kiểm thử, Mẫu Thiết kế, Tái cấu trúc mã nguồn, v.v..
- **Phát triển dựa trên giá trị:** dựa theo nguyên tắc “phần mềm chạy tốt là thước đo chính của tiến độ”, giúp nhóm luôn cố gắng để đạt được kết quả cuối và có thể bỏ đi những công việc dư thừa không trực tiếp đem lại giá trị cho sản phẩm.

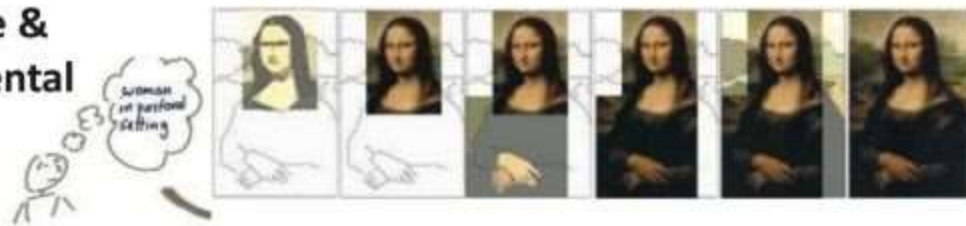
### Iterative



### Incremental



### Iterative & Incremental



**Hình 2.1.1 Tổng quan đặc điểm của Agile**

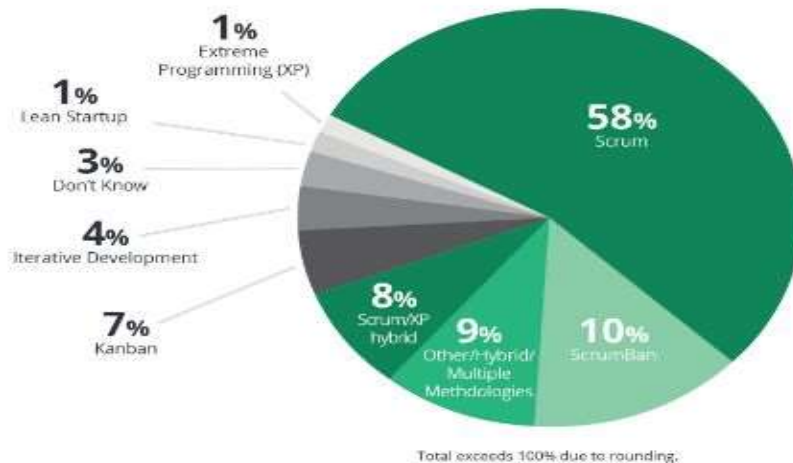
- **Ưu điểm:** đảm bảo rằng khách hàng liên tục tham gia trong mọi giai đoạn; các nhóm agile được tạo động lực và tự tổ chức để có khả năng cung cấp kết quả tốt hơn từ các dự án phát triển; phương pháp phát triển phần mềm nhanh đảm bảo rằng chất lượng của sự phát triển được duy trì; quá trình này hoàn toàn dựa trên tiến trình gia tăng, nên khách hàng và nhóm biết chính xác những gì được hoàn thành và những gì không. Điều này làm giảm rủi ro trong quá trình phát triển.
- **Nhược điểm:** không phải là phương pháp hữu ích cho các dự án phát triển nhỏ; đòi hỏi một chuyên gia để có những quyết định quan trọng trong cuộc họp; chi phí thực hiện một phương pháp nhanh hơn một chút so với các phương pháp phát triển khác; dự án có thể dễ dàng đi theo chiều hướng xấu nếu người quản lý dự án không rõ ràng kết quả họ muốn.

## 2.2 Các phương pháp của Agile:

### AGILE METHODS AND PRACTICES

#### AGILE METHODOLOGIES USED

Scrum and related variants continue to be the most common Agile methodologies used by respondents' organizations.



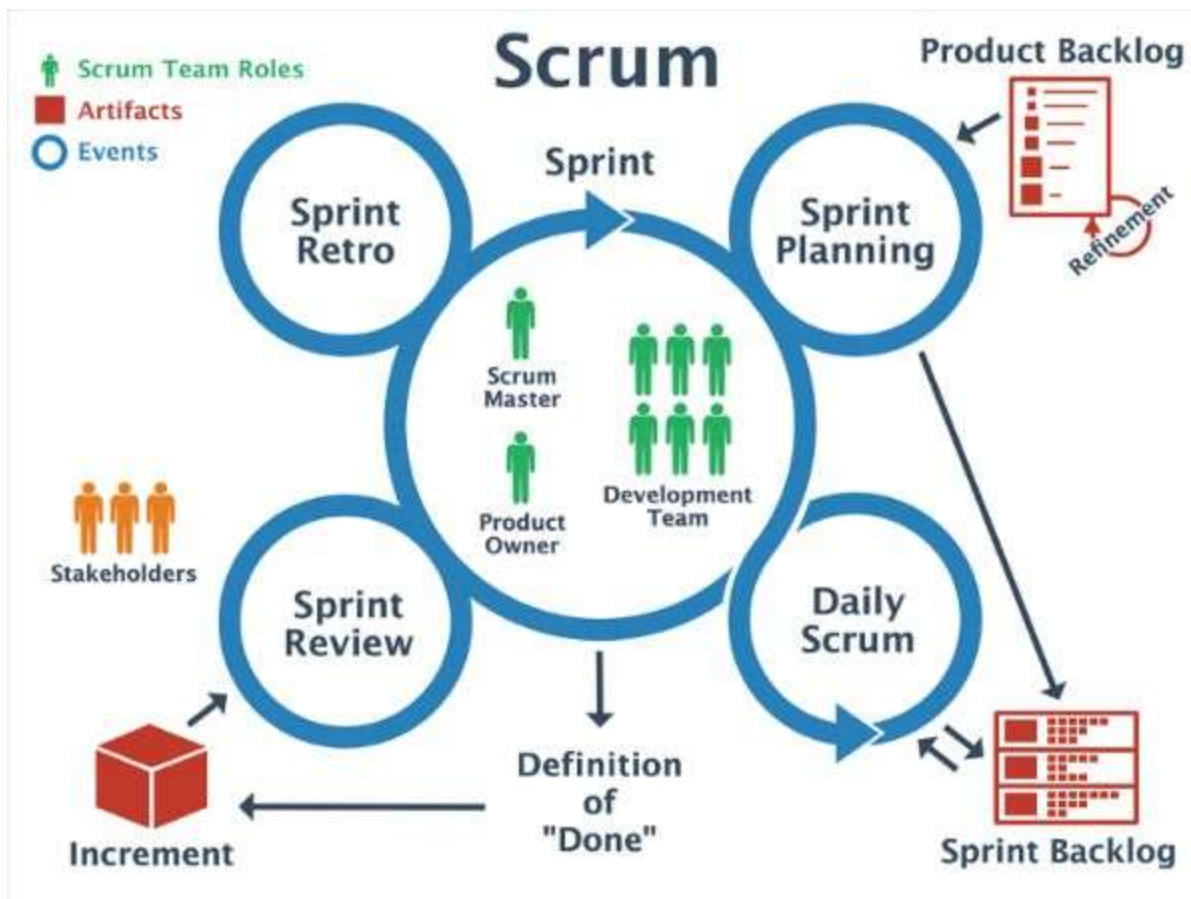
**Hình 2.2.** Khảo sát của VersionOne năm 2020, biểu đồ mô tả tỉ lệ áp dụng các phương pháp Agile

- Qua bảng thống kê trên ta có thể thấy phần đa các tổ chức doanh nghiệp ưa chuộng phương pháp Agile cụ thể là Scrum.
- Vậy Scrum là gì, cách thức hoạt động ra sao ? Và tại sao mô hình Scrum được lựa chọn rộng rãi như vậy? Và các phương pháp khác của Agile là gì, cách thức hoạt động ? Tại sao lại không phổ biến như Scrum ?

### 2.2.1 Scrum:

- Là một phương pháp Agile dùng cho phát triển sản phẩm , đặc biệt là phát triển phần mềm. Khung quản lý dự án được áp dụng rất rộng rãi, từ những dự án đơn giản với nhóm phát triển nhỏ cho đến những dự án có yêu cầu phức tạp với hàng trăm người tham gia, kể cả những dự án đòi hỏi khung thời gian cố định.
- Trong Scrum – Sprint là khoảng thời gian mà nhóm Scrum tiến hành tất cả các hoạt động cần thiết để sản xuất được một phần tăng trưởng.
- Scrum là một trong những phương pháp Agile quan trọng nhất sử dụng cơ chế lặp (iterative) và tăng trưởng (Incremental) để tối ưu hóa hiệu quả cũng như kiểm soát rủi ro.





**Hình 2.2.1.1 Tổng quan về phương pháp Scrum**

- Công việc được thực hiện bởi Nhóm **Scrum** thông qua từng phân đoạn lặp liên tiếp nhau được gọi là Sprint. Để hiểu được **Scrum** thì cần hiểu nguyên lý của **Scrum**, các Vai trò, Tạo tác, Sự kiện và sự vận hành của một vòng đời **Scrum**.
- Scrum có 3 trụ cột chính, 2 đặc điểm, 3 vai trò chủ chốt và 5 sự kiện của quy trình.
- Scrum có ba trụ cột chính là:
  - Tính minh bạch: thông tin liên quan tới quá trình phát triển phải minh bạch và thông suốt. Các thông tin đó là: tầm nhìn về sản phẩm, yêu cầu khách hàng, tiến độ công việc, các khúc mắc rào cản,... Để mọi người ở các vai trò khác nhau có đủ thông tin cần thiết để tiến hành các quyết định có giá trị nhằm nâng cao hiệu quả công việc. Các công cụ và cuộc họp trong Scrum luôn đảm bảo thông tin được minh bạch cho các bên.
  - Tính thanh tra: đảm bảo cho việc phát lộ các vấn đề cũng như giải pháp để thông tin đa dạng và hữu ích đến được với các bên tham gia quá trình phát

triển. Truy xét kỹ càng và liên tục là cơ chế khởi đầu cho việc thích nghi và các cải tiến liên tục trong Scrum.

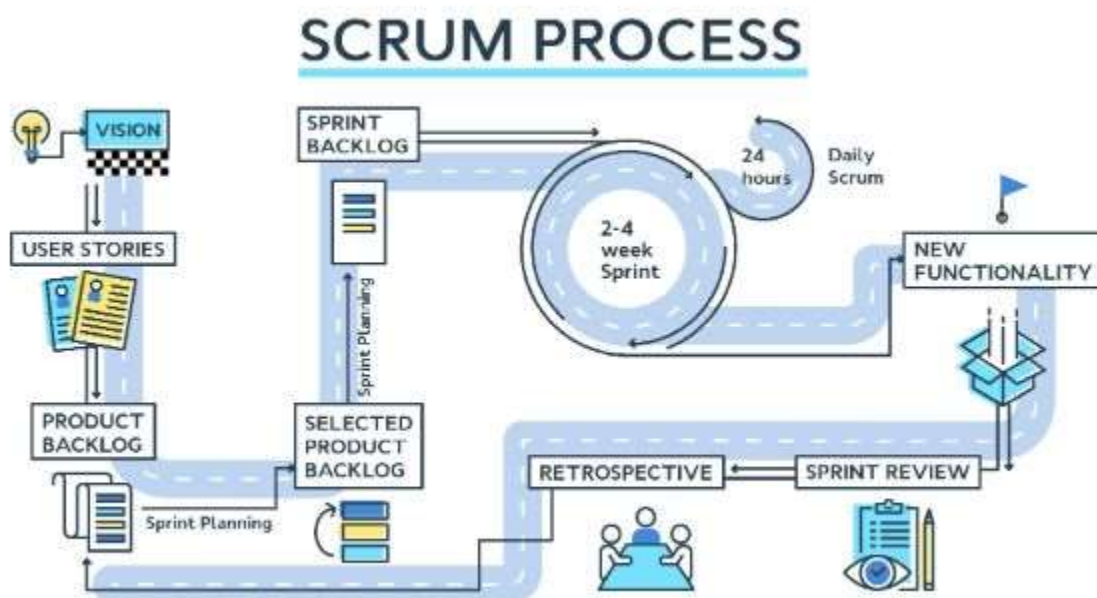
- Tính thích nghi: đảm bảo cho việc phát lộ các vấn đề cũng như giải pháp để thông tin đa dạng và hữu ích đến được với các bên tham gia quá trình phát triển. Truy xét kỹ càng và liên tục là cơ chế khởi đầu cho việc thích nghi và các cải tiến liên tục trong Scrum
- Nhóm Scrum có 2 đặc điểm: tự quản (self-managing) và liên chức năng (cross-functional):
  - Tự quản (self-managing): là thuật ngữ mới đối với self-organized. Là nhóm sẽ cùng ra quyết định sẽ làm gì, ai sẽ làm và làm như thế nào mà không phụ thuộc vào sự chỉ đạo của người khác. Nhắm tới mục tiêu chung là giúp tổ chức giải quyết các vấn đề phức tạp nhanh nhẹn và chất lượng hơn.
  - Liên chức năng (cross-functional): là nhóm liên chức năng bao gồm nhiều cá nhân với các chuyên môn khác nhau đủ năng lực được kết hợp lại cùng làm việc hướng tới một mục tiêu chung. Các cá nhân tham gia dự án có thể đến từ nhiều phòng ban chức năng khác nhau, cũng có thể xuất phát từ bên ngoài. Khi đã vào team thì phải làm việc tập trung cho đội như một đơn vị để hoàn tất mục tiêu chung.
- Trong Scrum có 3 vai trò là Product Owner (PO), Nhà phát triển, Scrum Master
  - Product Owner (PO): Vai trò này chịu trách nhiệm tối ưu hóa lợi nhuận trên đầu tư (ROI – Return On Investment) thông qua việc quyết định các tính năng của sản phẩm, đánh giá và sắp xếp độ ưu tiên của từng hạng mục, những hạng mục có độ ưu tiên cao thì sẽ được đưa vào phát triển trước, những hạng mục có độ ưu tiên thấp hơn thì sẽ được phát triển sau. Họ khác các giám đốc sản phẩm truyền thống là thay vì chỉ quản lý và ủy quyền cho người khác, các PO tham gia tích cực vào quá trình phát triển sản phẩm
  - Nhóm phát triển: là đội ngũ trực tiếp làm ra sản phẩm, họ bao gồm các chuyên gia có nhiệm vụ chuyển giao phần tăng trưởng ở cuối mỗi Sprint. Các Nhà phát triển không có sự phân chia các chức danh chuyên môn đặc thù cho từng thành viên, ví dụ như: kiểm thử viên, lập trình viên, chuyên gia thiết kế,

chuyên gia cơ sở dữ liệu,... mà tất cả đều được gọi chung là Nhà phát triển. Việc này giúp nâng cao tính sở hữu tập thể, trách nhiệm tập thể và bình đẳng giữa các thành viên.

- Scrum Master: Các Nhà phát triển không có sự phân chia các chức danh chuyên môn đặc thù cho từng thành viên, ví dụ như: kiểm thử viên, lập trình viên, chuyên gia thiết kế, chuyên gia cơ sở dữ liệu,... mà tất cả đều được gọi chung là Nhà phát triển. Việc này giúp nâng cao tính sở hữu tập thể, trách nhiệm tập thể và bình đẳng giữa các thành viên.
- Scrum có năm sự kiện: Sprint, Sprint planning, Daily Scrum, Sprint Retrospective
- Chi tiết các sự kiện của Scrum:
  - Sprint: là trái tim của Scrum, là khoảng thời gian cố định mà ở đó các nhà phát triển thực hiện công khai sản phẩm. Sprint được đóng khung thời gian không dài
  - Sprint Planning (lập kế hoạch): Là sự kiện diễn ra đầu Sprint để lên kế hoạch làm việc cho toàn bộ Sprint. Sự kiện này được chia làm 3 phần với 3 mục đích rõ ràng:
    - Phần 1: trả lời cho câu hỏi Why – “Tại sao chúng ta làm những thứ này” – do PO trình bày để xác định rõ Sprint Goal.
    - Phần 2: trả lời cho câu hỏi What - “Chúng ta sẽ hoàn thành những gì?”
      - sau khi trao đổi với PO, các Nhà phát triển sẽ lựa chọn những hạng mục từ Product Backlog để phát triển trong Sprint hiện tại.
    - Phần 3: trả lời câu hỏi How – “Chúng ta sẽ làm như thế nào?”. Đối với mỗi hạng mục được lựa chọn, các Nhà phát triển sẽ lên kế hoạch các công việc cụ thể để hoàn thành được mục tiêu Sprint (Sprint Goal).
- Với bộ 3 câu hỏi Why – What – How sẽ giúp Sprint tiến hành hiệu quả và rõ ràng hơn. Các Nhà Phát triển có quyền quyết định lựa chọn những hạng mục mà mình sẽ làm, không ai được phép can thiệp và gán công việc cho nhóm, kể cả Product Owner hay các lãnh đạo khác. Kết quả của buổi Lập kế hoạch Sprint là: Mục tiêu Sprint và Sprint Backlog.
  - Daily Scrum: Là buổi gặp mặt ngắn 15 phút hằng ngày của tất cả các thành viên Nhóm Phát triển để thanh tra và tái lập kế hoạch cho nhóm. Để giữ đơn

giản và tạo thói quen thì các buổi Scrum Hằng ngày phải diễn ra tại cùng một địa điểm vào cùng một khung thời gian. Scrum Master không bắt buộc tham dự nhưng phải đảm bảo Nhóm Phát triển đang thực hiện tốt sự kiện này.

- **Sprint review:** Là sự kiện diễn ra ở cuối Sprint nhằm thanh tra và thích nghi sản phẩm đang được xây dựng. Toàn bộ Nhóm **Scrum** (bao gồm Product Owner, Scrum Master và Nhóm Phát triển) tham dự sự kiện này. Sự kiện gồm 2 hoạt động chính đó là dùng thử sản phẩm và thảo luận về tình hình của sản phẩm, hướng đi tiếp theo và những điều chỉnh đối với sản phẩm nếu cần thiết. Product Backlog và Kế hoạch Phát hành có thể được điều chỉnh sau sự kiện này.
- **Sprint Retrospective (Sprint cải tiến):** Diễn ra sau sự kiện Sprint review nhằm thanh tra và thích nghi quy trình làm việc. Nói ngắn gọn, sự kiện này là để cải tiến cách làm việc. Nhóm Phát triển và Scrum Master bắt buộc tham gia sự kiện này. Product Owner có thể tham gia hoặc không. Kết quả của buổi làm việc này là một danh sách các thay đổi về cách làm việc được đưa vào áp dụng ngay trong Sprint tiếp theo.



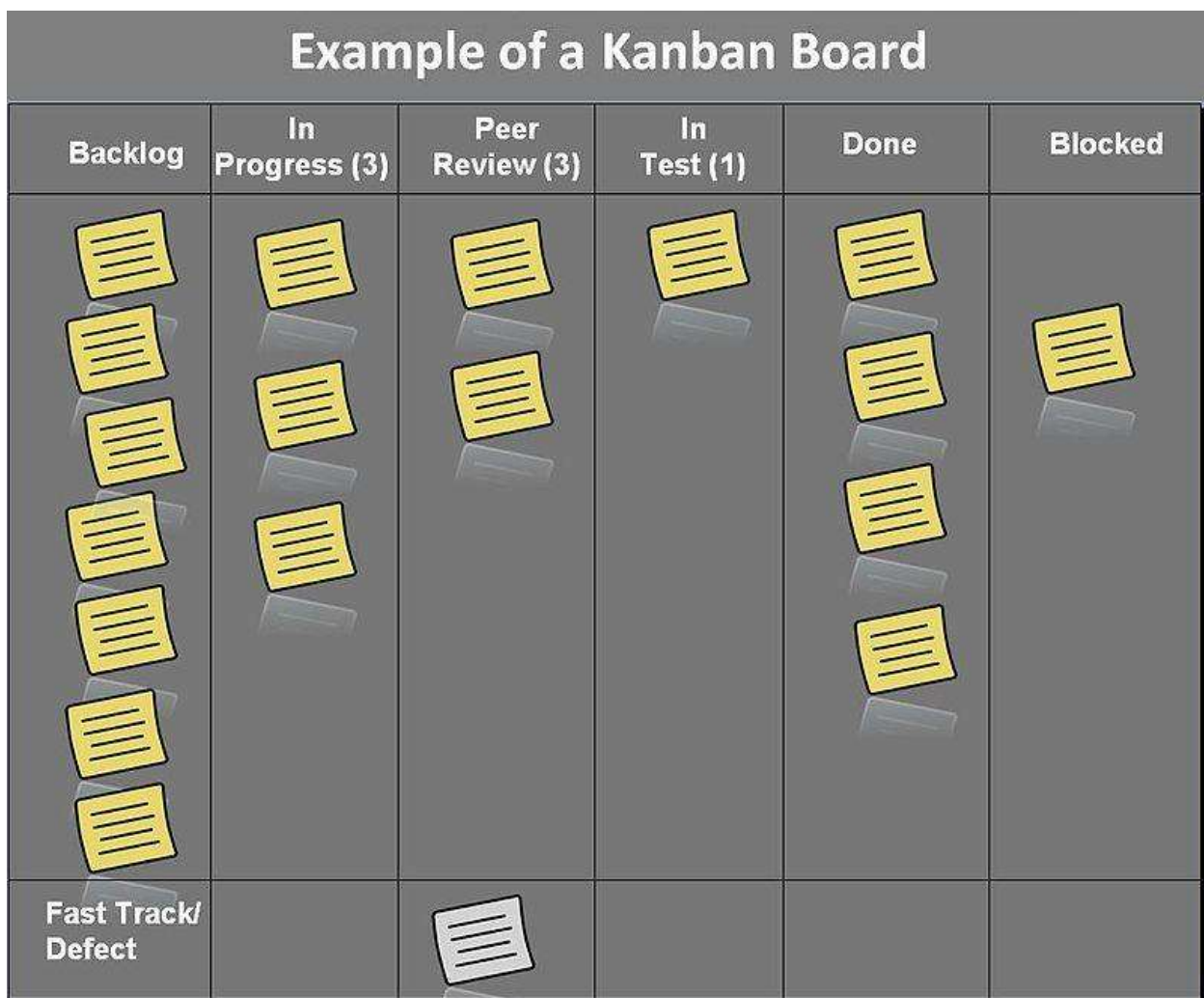
**Hình 2.2.1.2 Năm sự kiện trong Scrum**

- Các tạo tác trong Scrum – Scrum Artifact:

- **Product Backlog:** là nơi lưu trữ danh sách các tính năng mong muốn của sản phẩm và liên tục được cập nhật trong suốt vòng đời của sản phẩm. Mỗi PB sẽ được gắn với Product Goal. PO sẽ là người chịu trách nhiệm quản lý và bảo trì PB, bao gồm xác định nội dung, đánh giá độ ưu tiên và sắp xếp các hạng mục, làm mịn các hạng mục, làm rõ và giải thích tất cả mọi thắc mắc liên quan đến sản phẩm.
- **Sprint Backlog:** là bảng công việc được các Nhà Phát triển sử dụng để quản lý quá trình phát triển trong một Sprint và được cập nhật trong suốt Sprint. Chứa danh sách các hạng mục và các công việc cần làm tương ứng với từng hạng mục để hoàn thành mục tiêu của Sprint đó.
- **Incremental (Phần tăng trưởng):** là phần sản phẩm các Nhà Phát triển tạo ra cuối mỗi Sprint. là phần sản phẩm các Nhà Phát triển tạo ra cuối mỗi Sprint. Sprint không đơn giản tách quá trình phát triển thành các Sprint nhỏ liên tiếp nhau, mà cuối mỗi Sprint đòi hỏi nhóm phải chuyển giao một phần tính năng “hoàn chỉnh” của sản phẩm. oàn chỉnh ở đây được hiểu theo nghĩa được Product Owner chấp nhận dựa theo Định nghĩa Hoàn thành đã được thống nhất trước đó.
- Scrum mang lại cho người áp dụng các giá trị, lợi ích: sự tập trung, dũng cảm, cam kết, cởi mở, tôn trọng.
- Ngoài ra hiện nay với thời đại công nghệ phát triển các phần mềm giúp quản lý, hỗ trợ mô hình Scrum cũng được phát triển mạnh mẽ, phổ biến như: Trello, Jira, Asana,...

### 2.2.2 Kanban:

- Kanban (trong tiếng Nhật có nghĩa là bảng hiệu) là một phương pháp tinh gọn để quản lý và cải tiến công việc trên toàn hệ thống của con người. Cách tiếp cận này nhằm mục đích quản lý công việc bằng cách cân bằng nhu cầu công việc với năng lực hiện có và cải thiện việc xử lý vấn đề thắt cổ chai ở cấp hệ thống.



**Hình 2.2.2.1 Ví dụ về bảng Kanban**

- Phương pháp Kanban vận hành dựa trên 5 nguyên lý cốt lõi:
  - **Trực quan hóa luồng công việc:** Những dự án yêu cầu kiến thức hay như trong định nghĩa là sử dụng trí óc, có đặc điểm là không đo đếm được và vô hình. Do đó việc tìm ra cách để hình dung luồng công việc rất quan trọng trong việc tổ chức, tối ưu và theo dõi công việc.
  - **Giới hạn WIP (Work in Progress):** việc giới hạn số lượng các công việc làm đồng thời giúp cải thiện hiệu suất, tăng tính rõ ràng của các vấn đề và các trở ngại, tạo điều kiện cho sự cải tiến liên tục. Giúp nhóm dễ dàng hơn trong việc xác định vấn đề và giảm thiểu chi phí và sức lực cần có để thay đổi, tạo động lực “kéo” – pull ổn định trong công việc thông qua nỗ

lực phát triển, vì tác vụ mới chỉ có thể được tiến hành khi các tác vụ hiện tại đã được hoàn thành.

- **Quản lý luồng công việc:** việc theo dõi luồng công việc thông qua một hệ thống, vấn đề có thể được xác định và các phương án thay đổi sẽ được đánh giá dựa trên mức độ hiệu quả. Giải thích quy trình công việc rõ ràng rất quan trọng để nhóm có thể trao đổi, bàn luận về các cải tiến, cải thiện một cách công khai, khách quan thay vì theo định hướng chủ quan và cảm tính.
- **Cùng nhau tiến bộ:** Qua các thước và thử nghiệm khoa học, nhóm nên nắm vững và cải thiện một cách tập thể các quy trình mà nhóm đang sử dụng.

#### 2.2.2.1 So sánh 2 phương pháp Scrum – Kanban:

- Sơ lược lại về 2 phương pháp:
  - + **Scrum** là một framework để phát triển bền vững các sản phẩm phức tạp, sử dụng cơ chế lặp và tăng trưởng để tối ưu hóa hiệu quả cũng như kiểm soát rủi ro. Mang lại giá trị kinh doanh trong thời gian ngắn.
  - + **Kanban** là một hệ thống trực quan để quản lý công việc. Nó hình dung cả quá trình và công việc thực tế trong quá trình đó.
- So sánh hai phương pháp:
  - + Giống nhau: cả 2 phương pháp đều chia nhỏ các task lớn và phức tạp thành những đoạn nhỏ và hoàn thành theo một quy trình nhất định, đều cải tiến liên tục, tối ưu hóa công việc và quá trình, tập trung vào dòng chảy công việc để khuyến khích các thành viên tham gia vào quy trình.

+ Khác nhau:

SCRUM	KANBAN
+ Công việc hay nhiệm vụ được hoàn thành trong thời gian hoạch định của sprints, thường là trong 2-4 tuần. Mục tiêu là hoàn thành sản phẩm để chuyển giao cho khách hàng.	+ Không có sprints cố định. Thay vào đó các nhóm sẽ lấy nhiệm vụ từ danh sách tồn đọng và ưu tiên những việc cần phải hoàn thành.
+ Sản phẩm được released theo nhịp cụ thể, được xác định theo độ dài của sprints. Nên một nhóm có thể release sản phẩm sau mỗi 3 sprints hay mỗi 6 tuần.	+ Sản phẩm được release liên tục hoặc bất cứ khi nào có sản phẩm chuyển giao được tạo ra.
+ Tất cả các thành viên trong nhóm đều phải tham gia vào dự án. Thường có 3 bên quản lý PO, Scrum master và nhóm phát triển. Các bên đều phải làm việc cùng nhau, để đảm bảo cân bằng giữa yêu cầu và sản phẩm cuối.	+ Mỗi thành viên có thể làm việc riêng lẻ theo chuyên môn của mình, nhưng có quá nhiều việc chuyên môn hóa sẽ làm ảnh hưởng hiệu quả làm việc của nhóm.
+ Bắt đầu sprint, báo cáo hàng ngày, đánh giá sprint và cải tiến sprint; đó là các quy trình quan trọng của Scrum.	+ Có sự tập trung chủ yếu về sự cải tiến của quy trình, nhưng không có các cuộc họp theo tiêu chuẩn.

- Có một số câu hỏi được đặt ra cho phương pháp Scrum và Kanban về việc phương pháp nào tốt hơn: thì qua sự so sánh ở trên ta thấy được cả hai phương pháp đều có điểm mạnh, điểm yếu nhất định. Vậy việc tìm ra phương pháp tốt nhất cho công ty doanh nghiệp và



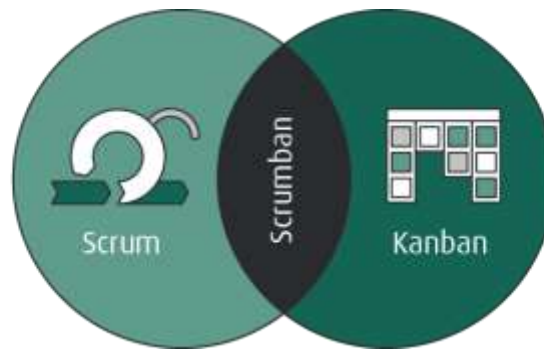
phát triển dự án còn tùy thuộc vào nhu cầu, điều kiện và chiến lược riêng của từng doanh nghiệp để lựa chọn cho mình.

- Ngoài ra việc kết hợp các ưu điểm của hai phương pháp đã tạo ra Scrumban.

### 2.2.3 Scrumban:

- Là sự kết hợp của Scrum – giải pháp tốt nhất cho product và phát triển project và Kanban – giải pháp tốt nhất hỗ trợ sản xuất. Bao gồm toàn bộ những ưu điểm của 2 phương thức trên, khuyến khích các đội phải liên tục cải tiến quy trình cùng với phương thức Kanban.

Project Management  
Methodologies



- Scrumban được giới thiệu như một quy trình đơn giản để quản lý những project phức tạp. Hiện nay Scrumban được áp dụng tốt nhất khi phát triển trên web, software development hoặc maintenance.
- Sử dụng quy tắc tự nhiên của Scrum là Agile, sử dụng quy trình phát triển của Kanban để team có thể tiếp tục cải tiến quy trình. Tạo ra Scrumban, chúng ta có thể lập những kế hoạch lặp lại trong những khoảng thời gian đều đặn, đồng bộ với việc đánh giá và có thể xem lại những gì đã làm. Điều này làm giảm đáng kể chi phí.
- Lợi ích:

- Chất lượng
- Đúng thời gian
- Thời gian quản lý ngắn
- Tiếp tục cải tiến
- Ít rủi ro, giảm lãng phí
- Có thể thêm chức năng vào Scrum khi cần

#### **Hình 2.2.3.1 Lợi ích của Scrumban mang lại**

- Scrumban nên được sử dụng khi nào:
  - Khi dự án bảo trì.
  - Dự án mà thường xuyên có lỗi không mong đợi.

#### **2.2.4 Feature-Driven Development (FDD):**

- Là phương pháp luận tập trung vào việc xây dựng và thiết kế các tính năng của hệ thống. Các nhóm phát triển hệ thống làm việc trong các giai đoạn ngắn để phát triển các chức năng của hệ thống. Nói một cách đơn giản, FDD tập trung vào việc phát triển các hệ thống theo tính năng cụ thể.
- FDD được phát triển để đáp ứng nhu cầu cụ thể của dự án phát triển phần mềm lớn. Với các công ty lớn và dự án quy mô thì FDD là phương pháp phù hợp, phương pháp này phụ thuộc vào chief developers và áp dụng phương pháp Top-down ngược lại với phương pháp Agile khác như XP.

## Feature-Driven Development In A Nutshell

Feature-Driven Development is a pragmatic software process that is client and architecture-centric. Feature-Driven Development (FDD) is an agile software development model that organizes workflow according to which features need to be developed next.



### 2.2.5 Crystal:

- là một phương pháp phát triển hệ thống linh hoạt khác với ba khái niệm cơ bản:
  - Chartering – giống như giai đoạn yêu cầu trong đó các hoạt động liên quan được phác thảo. Điều này liên quan đến các nghiên cứu khả thi, phát triển kế hoạch và cụ thể hóa phương pháp luận.
  - Cyclic delivery (Phát hành theo chu kỳ) – giai đoạn này chứa hai hoặc nhiều chu kỳ phát hành. Tại đây, nhóm sẽ tinh chỉnh các kế hoạch, thực hiện các yêu cầu của tập hợp con, xem xét kế hoạch dự án và áp dụng phương pháp luận.
  - Wrap-up – giai đoạn cuối cùng thường liên quan đến việc triển khai hệ thống tới máy chủ và người dùng. Một phân tích sau khi triển khai cũng được thực hiện.



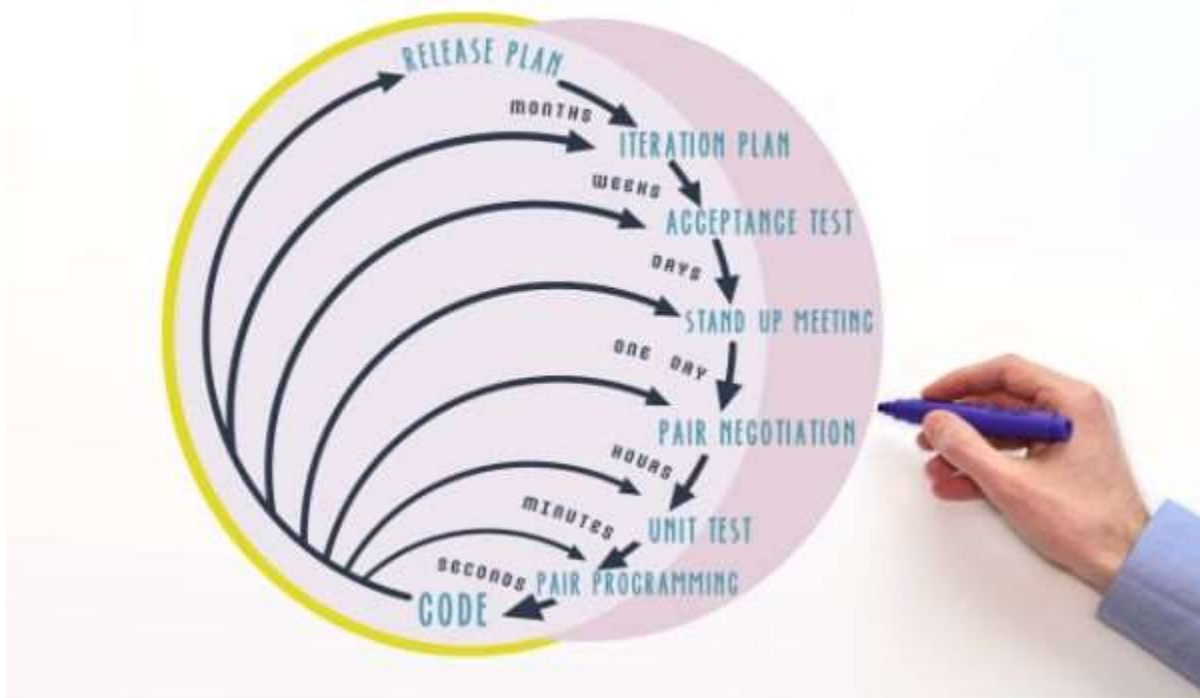
**Hình 2.2.5.1 Hình mô tả phương pháp Crystal**

- Phương pháp Crystal tập trung vào các nguyên tắc như: Con người (People), Interactions (Sự tương tác), Community (Cộng đồng), Skills (Kỹ năng), Talent (Tài năng) và Communication (Giao tiếp). Mục đích của Crystal là đưa ra một quy trình phát triển phần mềm tốt nhất có thể. Theo đó, vấn đề cốt lõi phải là sự tương tác và cộng sinh giữa những người liên quan đến dự án.

## **2.2.6 Extreme Programming:**

- Extreme Programming (XP): là một phương pháp phát triển phần mềm hướng đến việc nâng cao chất lượng phần mềm và khả năng đáp ứng với thay đổi yêu cầu người dùng. Chủ trương đưa ra các bản phát hành thường xuyên thông qua các chu trình phát triển ngắn, nhằm nâng cao năng suất và tạo ra các sprint để tiếp nhận các yêu cầu của người dùng mới.

# EXTREME PROGRAMMING



Hình 2.2.3 Tổng quan về Extreme Programming (XP)

- XP tập trung tập trung vào việc áp dụng tốt nhất những kỹ thuật lập trình, giao tiếp rõ ràng và làm việc nhóm để tạo những sản phẩm tốt nhất.
- Một số thành phần và đặc điểm của **XP**: Lập trình cặp, Rà soát mã nguồn, Kiểm thử đơn vị, Giữ mã nguồn đơn giản và rõ ràng, Sẵn sàng đón nhận các thay đổi, Trao đổi thường xuyên với khách hàng, Trao đổi thường xuyên giữa các nhà phát triển.
- Giá trị của XP: Giao tiếp, đơn giản, phản hồi, phản hồi từ hệ thống, phản hồi từ khách hàng, phản hồi từ nhóm, tinh thần dũng cảm.
- XP có 6 vai trò chính:
  - + Tracker: đóng vai trò theo dõi tiến độ dự án & phát hiện các vấn đề mà developer gặp phải. Tracker trò chuyện với từng developer 1-2 lần/tuần, trao đổi với họ về tiến độ & lắng nghe khó khăn developer chia sẻ. Khi phát hiện có vấn đề ảnh hưởng tiến độ, Tracker có thể tiếp cận Customer, Coach, hay các developer khác để tìm giải pháp xử lý.

- + Customer: tương tự Product Owner trong Scrum, Customer là người quản lý product backlog, viết User Story, đặt thứ tự ưu tiên cho backlog item.
- + Developer: là người tiếp nhận backlog item từ Customer & triển khai các backlog item
- + Tester: là người phụ trách công việc kiểm thử, đảm bảo chất lượng sản phẩm
- + Coach: là người theo dõi quá trình làm việc của development team, quản lý về con người & chất lượng, hướng dẫn thành viên làm theo các programming practices của XP.
- + “Doomsayer”: có thể là 1 thành viên của XP, hoặc một đặc điểm; bất cứ thành viên ở vị trí nào trong XP nếu có khả năng phát hiện, theo dõi, cảnh báo rủi ro đều có thể gọi là "Doomsayer"

#### 2.2.6.1 So sánh Scrum và Extreme Programming (XP):

- Giống nhau:
  - + Cả Scrum và Extreme Programming đều hướng tới việc xây dựng sản phẩm với release cycle ngắn hơn, chất lượng tốt hơn và tăng khả năng tự quản lý
  - + Đều phát triển dựa trên các sprint, có độ dài cố định, có sprint backlog, sprint goal
  - + Các hoạt động của Agile như daily meeting, sprint planning, sprint retro, sprint review.
- Khác nhau:

Yếu tố so sánh	SCRUM	XP
Độ dài sprint	+ Kéo dài từ 1-4 tuần	+ Chỉ kéo dài từ 1-2 tuần. Vì XP chú trọng việc release nhanh và liên tục
Khả năng thay đổi sprint backlog	+ Hạn chế tối đa các thay đổi xảy ra khi đã kết thúc sprint planning và goal đã đề ra	+ Có thể thay đổi thêm/bớt miễn là Dev team chưa làm tới + backlog item được thêm mới phải có độ lớn tương đương với backlog item cũ.

Độ ưu tiên của User Story ảnh hưởng tới thứ tự thực hiện	+ PO đưa ra độ ưu tiên của từng User Story, nhưng Dev team sẽ tự quyết định thứ tự thực hiện trong sprint miễn đạt được sprint Goal	+ Thứ tự phát triển phải theo đúng thứ tự ưu tiên của User Story do PO/khách hàng yêu cầu.
Áp dụng Programming practices	+ Không yêu cầu bắt buộc Dev team phải tuân theo bất cứ programming practices cụ thể nào	+ Yêu cầu chặt chẽ về việc tuân thủ programming practices để đảm bảo chất lượng sản phẩm và khả năng deliver liên tục (Continuous Delivery )
Roles trong framework	+ Có 3 vị trí chính : Product Owner, Scrum Master và Nhà phát triển	+ Có 6 vị trí: Tracker, Customer, Developer, Coach, Tester, “Doomsayer”

### 2.3 Mô hình thác nước Waterfall:

- Được biết đến là một trong những mô hình quản lý dự án dễ hiểu nhất hiện nay. Là một phương pháp quản lý dự án dựa trên quy trình thiết kế tuần tự và liên tiếp.
- Trong mô hình Waterfall, các giai đoạn của dự án được thực hiện lần lượt và nối tiếp nhau. Giai đoạn mới chỉ được bắt đầu khi giai đoạn trước nó đã được hoàn thành.
- Một mô hình Waterfall đơn giản có 6 giai đoạn:
  - Giai đoạn yêu cầu (Requirement Analysis): nhóm thực hiện tìm kiếm các yêu cầu liên quan đến dự án như: xác định dự án sẽ giải quyết nhu cầu kinh doanh nào, yêu cầu của người dùng đối với sản phẩm được phát triển bởi dự án, các ràng buộc và rủi ro đi kèm.
  - Giai đoạn thiết kế (Design): Nhóm tạo ra thiết kế cho sản phẩm để giải quyết mọi yêu cầu, ràng buộc và mục tiêu thiết kế, bản thiết kế điển hình cần được hoàn thành càng cụ thể càng tốt.

- Giai đoạn xây dựng (Development): Sản phẩm được chế tạo để hỗ trợ thiết kế. Đôi khi, sản phẩm được xây dựng trong các đơn vị dùng để thử nghiệm và tích hợp trong giai đoạn tiếp theo.
  - Giai đoạn kiểm chứng (Test): Các bộ phận của sản phẩm được kiểm tra. Nếu cần sẽ được tích hợp lại với nhau để thử nghiệm và toàn bộ hệ thống được kiểm tra để tìm ra lỗi và đảm bảo các mục tiêu thiết kế.
  - Giai đoạn triển khai (Deployment): Sản phẩm được thử nghiệm thực sự đi vào hoạt động. Đối với các dự án thuộc lĩnh vực công nghệ thông tin, sản phẩm được triển khai vào môi trường để người dùng có thể bắt đầu sử dụng nó.
  - Giai đoạn bảo trì (Maintenance): Là một khoảng thời gian giám sát ngắn. Trong đó nhóm dự án giải quyết các vấn đề của khách hàng.
- **Ưu điểm:** thích nghi tốt với những nhóm linh hoạt, áp đặt một tổ chức có kết cấu chặt chẽ, cho phép những thay đổi thiết kế sớm, thích hợp cho những dự án theo hướng đến mốc.
  - **Nhược điểm:** không phải là một mô hình lý tưởng cho một dự án kích thước lớn, nếu yêu cầu không rõ ràng ngay từ đầu thì đó là phương pháp kém hiệu quả hơn, rất khó di chuyển trở lại cái giai đoạn trước đó để thay đổi, quá trình thử nghiệm bắt đầu khi quá trình phát triển kết thúc do đó nó có nguy cơ cao của các lỗi được tìm thấy sau giai đoạn phát triển, và rất tốn kém để sửa các lỗi.
  - **Khi nào nên áp dụng mô hình:** nắm vững được công nghệ phát triển của công nghệ, loại bỏ những yêu cầu mập mờ, không rõ ràng, có lượng tài nguyên phát triển phong phú và trình độ chuyên môn, kỹ thuật cao, có thể phù hợp cho dự án nhỏ, ngắn hạn.
  - vì sao lại ưu tiên sử dụng mô hình Agile hơn Waterfall ?
  - **Agile** là triết lý với các phương pháp mới thay thế cho phương pháp của mô hình Waterfall truyền thống đã khẳng định vị thế khi đem đến cho cá nhân và tổ chức

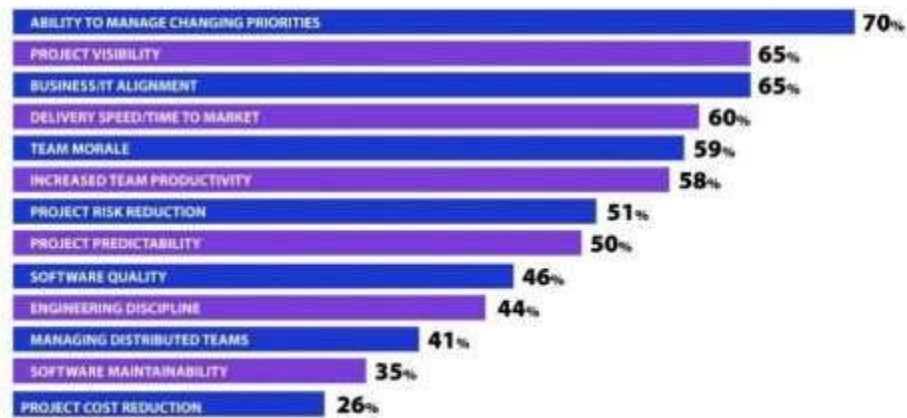


những lợi ích nhất định. Theo khảo sát VersionOne năm 2020 về việc triển khai Agile đã cho thấy có sự cải tiến trong các lĩnh vực sau:

## BENEFITS OF AGILE

### BENEFITS OF ADOPTING AGILE

We continue to see many benefits realized by companies adopting Agile. The theme of the top 5 reported benefits is speed and adaptability. This corresponds with the top reported reasons for adopting Agile.



- Với báo cáo CHAOS của Standish Group năm 2015 đã cho thấy các dự án **Agile** so với các dự án truyền thống (Waterfall) có tỷ lệ thành công cao hơn 3 lần. Cụ thể trong bảng dưới đây:

Quy mô dự án	Phương pháp	Thành công	Thử thách	Thất bại
Tổng kết	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Lớn	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Vừa	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Nhỏ	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

- Ngoài ra việc áp dụng mô hình Agile đem lại một số lợi ích như: sử dụng linh hoạt, sáng tạo và cho năng suất cao hơn.

#### 2.2.4 So sánh hai mô hình Agile và Waterfall:

- Giống nhau: cả hai mô hình đều có mục tiêu chung, đều nhằm mục đích tạo ra và chuyển giao những sản phẩm có chất lượng bằng cách hiệu quả nhất.
- Khác nhau:

AGILE	WATERFALL
+ Tách vòng đời phát triển dự án thành các Sprint hay task nhỏ.	+ Quá trình phát triển phần mềm được chia thành các giai đoạn riêng biệt.
+ Thực hiện theo một cách tiếp cận gia tăng	+ Là một quá trình thiết kế tuần tự.
+ Phương pháp nhanh được biết đến với tính linh hoạt của nó.	+ Thác là một phương pháp phát triển phần mềm có cấu trúc nên hầu hết thời gian nó có thể khá cứng nhắc.
+ Agile có thể được coi là một bộ sưu tập của nhiều dự án khác nhau.	+ Phát triển phần mềm sẽ được hoàn thành như một dự án duy nhất.
+ Agile là một phương pháp khá linh hoạt cho phép thay đổi được thực hiện trong các yêu cầu phát triển dự án ngay cả khi kế hoạch ban đầu đã được hoàn thành.	+ Không có phạm vi thay đổi các yêu cầu khi phát triển dự án bắt đầu.
+ Phương pháp nhanh, theo một cách tiếp cận phát triển lặp lại vì quy hoạch, phát triển, tạo mẫu và các giai đoạn phát triển phần mềm khác có thể xuất hiện nhiều lần.	+ Tất cả các giai đoạn phát triển dự án như thiết kế, phát triển, thử nghiệm, vv được hoàn thành một lần trong mô hình Thác
+ Kế hoạch kiểm tra được xem xét sau mỗi lần chạy nước rút	+ Kế hoạch kiểm tra hiếm khi được thảo luận trong giai đoạn thử nghiệm.

+ Phát triển nhanh là một quá trình trong đó các yêu cầu được dự kiến sẽ thay đổi và phát triển.	+ Phương pháp này là lý tưởng cho các dự án có yêu cầu nhất định và thay đổi không được mong đợi.
+ Trong phương pháp Agile, thử nghiệm được thực hiện đồng thời với phát triển phần mềm.	+ Trong phương pháp này, giai đoạn "Thử nghiệm" xuất hiện sau giai đoạn "Xây dựng"
+ Mô tả chi tiết dự án có thể được thay đổi bất cứ lúc nào trong quá trình SDLC (Software Development Life Cycle).	+ Mô tả chi tiết cần thực hiện phương pháp tiếp cận phát triển phần mềm thác nước.
+ Các thành viên của Nhóm Agile có thể hoán đổi cho nhau, do đó, chúng hoạt động nhanh hơn. Cũng không cần thiết cho các nhà quản lý dự án vì các dự án được quản lý bởi toàn bộ nhóm	+ Trong phương pháp thác nước, quy trình luôn đơn giản như vậy, người quản lý dự án đóng một vai trò thiết yếu trong mọi giai đoạn của SDLC.

- **Kết luận:** Agile và Waterfall là các phương pháp phát triển phần mềm rất khác nhau và tốt theo cách tương ứng.

## **CHƯƠNG 3: KẾT LUẬN VÀ KIẾN NGHỊ**

### **3.1 Kết luận:**

Về mặt lý thuyết, trong quá trình thực hiện đồ án em đã tìm hiểu về các kiến thức phát triển phần mềm, tìm hiểu về các phương pháp, các mô hình theo mục tiêu đề ra.

Về ứng dụng, mặc dù là một chủ đề không quá xa lạ, tuy nhiên vẫn có nhiều phương pháp áp dụng, các mô hình phát triển phần mềm được nghiên cứu rộng rãi cả trong và ngoài nước để phục vụ cho nhu cầu phát triển của các công ty, doanh nghiệp. Đây là vấn đề có ý nghĩa quan trọng đối với ngành công nghệ phát triển phần mềm..

Về các mặt khác, qua đồ án thực tập chuyên môn em đã học hỏi và phát triển được nhiều kỹ năng mềm như kỹ năng tìm hiểu, đọc báo khoa học, kỹ năng tổng hợp kiến thức cũng như kỹ năng phân tích dữ liệu thu thập.

### **3.2 Tồn tại:**

Do thời gian nghiên cứu và trình độ của em còn hạn hẹp nên việc nghiên cứu còn nhiều hạn chế. Các kiến thức về công nghệ phát triển phần mềm, mô hình ứng dụng chưa tìm hiểu chuyên sâu. Do các tác nhân liên quan đến nguồn kinh phí huấn luyện mô hình hạn chế, tài nguyên máy tính còn hạn hẹp nên mô hình chưa được hoàn thiện tốt nhất. Xây dựng mô hình còn nhiều khó khăn và chưa phát triển chưa hoàn thiện, còn gặp nhiều hạn chế về mô hình.

### **3.3 Kiến nghị:**

Với những mặt tồn tại, nếu có nhiều thời gian và tài nguyên hệ thống máy tính hơn em sẽ cố gắng hoàn thiện nghiên cứu mô hình phát triển phần mềm hơn. Được tham gia trực tiếp vào dự án phát triển phần mềm để được tìm hiểu chuyên sâu các mô hình khác nhau. Teamwork thiết kế dự án ổn áp hơn cho đề tài nghiên cứu.

