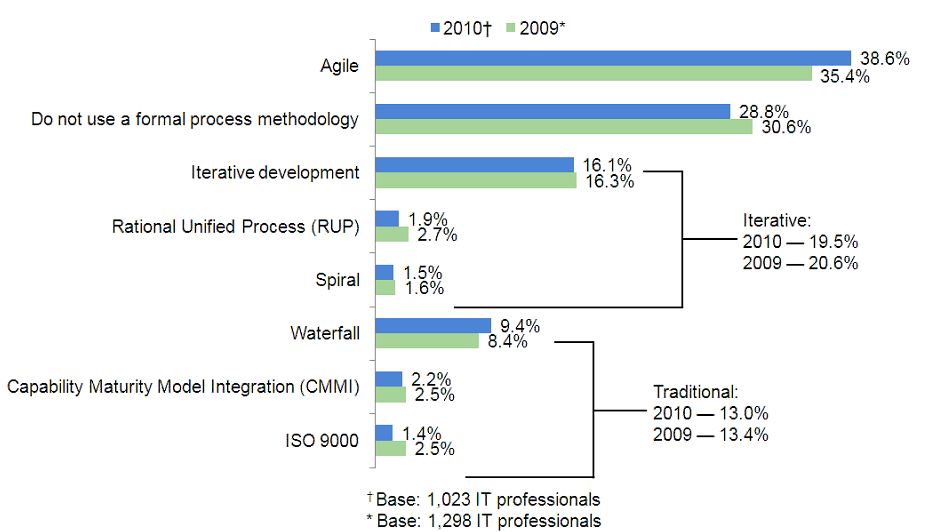
# Giới thiệu

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 như thác nước hay CMMI (xem Hình 1).



*Hình 1: Mức độ phổ biến của các phương pháp (Nguồn: Forrester Research)*

## Tuyên ngôn Agile (Agile Manifesto)

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.*

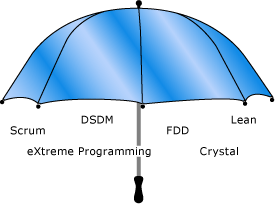
Bên cạnh đó, các nhà phát triển còn nhấn mạnh mười hai nguyên lý sau Tuyên ngôn Agile nhằm giúp các nhà phát triển vận dụng các phương pháp agile trong thực tiễn. Các nguyên lý được liệt kê sau đây:

* 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.

## Phân loại Agile

Phát triển linh hoạt bản thân nó không phải là một phương pháp. Nó là một thuật ngữ chung mô tả rất nhiều các phương pháp linh hoạt. Tại thời điểm ký kết Tuyên ngôn Agile năm 2001, những phương pháp này bao gồm: Scrum, XP, Crystal, FDD, và DSDM. Kể từ thời điểm đó, Lean cũng đã nổi lên như là một phương pháp linh hoạt có giá trị do vậy cũng được đưa vào trong chiếc ô của các phương pháp Agile trong hình minh họa dưới đây.



Mỗi phương pháp phát triển linh hoạt có một cách tiếp cận hơi khác nhau để thực hiện các giá trị cốt lõi trong tuyên ngôn Agile, cũng giống như các ngôn ngữ máy tính có các biểu hiện tính năng cốt lõi trong lập trình hướng đối tượng theo những cách khác nhau. Một cuộc khảo sát gần đây cho thấy rằng, khoảng 50% học viên theo học phương pháp phát triển linh hoạt nói rằng đội của họ đang làm Scrum, 20% khác nói rằng họ đang làm Scrum với các thành phần của XP. Khoảng 12% nói rằng họ chỉ áp dụng XP. Do có hơn 80% áp dụng phát triển linh hoạt trên toàn thế giới là sử dụng Scrum và XP, nên bản MSF cho phát triển phần mềm linh hoạt phiên bản 5.0 tập trung vào quy trình cốt lõi và thực tiễn áp dụng của Scrum và XP.

## 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.

Tất cả các nhóm agile phải xác lập những gì họ muốn nói là "phần mềm chạy tốt", cái thường được biết như là định nghĩa hoàn thành. Ở mức độ cao, một phần của chức năng hoàn thành chỉ khi các tính năng của chúng vượt qua tất cả các kiểm thử và có thể được vận hành bởi người dùng cuối. Ở mức thấp nhất, các nhóm phải vượt qua được kiểm thử đơn vị (unit test) và kiểm thử hệ thống. Các nhóm tốt nhất còn bao gồm việc kiểm thử tích hợp, kiểm thử hiệu năng, và kiểm thử chấp nhận của khách hàng trong định nghĩa hoàn thành đối với một phần chức năng. Thông qua nguồn dữ liệu phong phú từ các dự án, một công ty CMMI Cấp độ 5 cho thấy việc xác định kiểm thử chấp nhận cùng với các tính năng, triển khai một loạt các tính năng và theo độ ưu tiên, ngay lập tức chạy các kiểm thử chấp nhận với mỗi tính năng, và sửa bất cứ một lỗi nào có độ ưu tiên cao nhất sẽ tăng gấp đôi tốc độ sản xuất và giảm các sai sót đến 40%. Điều này có được từ một công ty có tỷ lệ sai sót thấp nhất thế giới.

Tuyên ngôn Agile khuyến nghị các nhóm cung cấp phần mềm chạy tốt sau một khoảng thời gian nhất định. Đồng thuận với định nghĩa hoàn thành là một trong những cách thực tế để nhóm agile mang lại hiệu suất và chất lượng cao, cái cần thiết để hoàn thành mục tiêu này.

### 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 hoạt động 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 thành công.

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 (chủ sản phẩm), 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. Chủ sản phẩm 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.

Một nhóm XP đầu tiên đã bắt đầu với một dự án CNTT nội bộ. Họ có thể có sẵn người sử dụng đầu cuối của công ty trong nhóm làm việc với họ hàng ngày. Khoảng 10% thời gian, các tư vấn viên và nhóm nội bộ có thể tìm được một người dùng cuối có thể làm việc với nhóm từng ngày. 90% thời gian còn lại, họ phải cử ra người đại diện cho khách hàng. Người này, được nhóm XP gọi là Customer (khách hàng), làm việc trực tiếp với người dùng cuối để cung cấp một danh sách các tính năng rõ ràng cùng độ ưu tiên cho phép đội phát triển có thể thực hiện.

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 dữ liệu trong ngành công nghiệp cho thấy rằng các dự án linh hoạt 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 triển linh hoạt đã nhận ra điều đó, và do vậy, chúng đã tạo ra một vị trí đặc biệt trong đội hình 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 khi sản phẩm đang được phát triển.

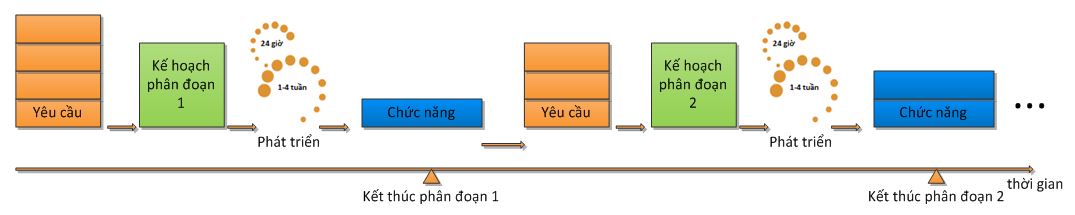
Tất cả các phương pháp phát triển linh hoạt đều được tích hợp sẵn những tiến trình thay đổi các kế hoạch trong một khoảng thời gian đều đặn dựa trên những thông tin phản hồi từ phía khách hàng cũng như bên đại diện của khách hàng. Các kế hoạch được thiết kế để sao cho luôn cung cấp giá trị kinh doanh 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. Kết quả là, khách hàng thì vui vẻ hơn, và các nhà phát triển thì thích thú với công việc của họ hơn. Các phương pháp phát triển linh hoạt dựa trên những hiểu biết đó, để thành công hơn chúng phải có kế hoạch để thay đổi. Đó là lý do tại sao chúng thiết lập các quy trình, chẳng hạn như Sơ kết và Cải tiến, được thiết kế đặc biệt để thay đổi các ưu tiên thường xuyên dựa trên thông tin phản hồi của khách hàng và giá trị kinh doanh.

## Đặc trưng 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 (Iterative)***

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 mục tiêu 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. Có phương pháp như Scrum thậm chí sử dụng phương pháp lập kế hoạch just-in-time trong quá trình phát triển. 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 2: Các phân đoạn lặp đi lặp lại trong agile*

***Tính tiệm tiến (Incremental) và tiến hóa (Evolutionary)***

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 (potentially shippable product increment of functionality). 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 (hay thích nghi – adaptive)***

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 v.v.) 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, chủ sản phẩm (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. Theo 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(cross-functionality) và tự tổ chức(self-organizing). Theo đó, các nhóm này tự thực hiện lấy việc phân công công việc mà không dựa trên các mô tả cứng về chức danh (title) 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" (command and control). Nhóm tự tổ chức có nghĩa là nó đã đủ các kĩ năng (competency) 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 lý tiến trình thực nghiệm (Empirical Process Control)***

Các nhóm agile ra các quyết định dựa trên các dữ liệu thực tiễn thay vì tính toán lý thuyết hay các tiền giả định (prescription). 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 (short feedback life cycle) 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(face-to-face communication)***

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 trở lực để thực hiện thành công mục tiêu của dự án.

***Phát triển dựa trên giá trị (value-based development)***

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 chủ sản phẩm 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.

# Các phương pháp Agile

## Scrum

### Giới thiệu Scrum

Scrum (n) : Một mô hình mà trong đó mọi người có thể giải quyết vấn đề thích nghi phức tạp , trong khi cung cấp sản phẩm hiệu quả và sáng tạo với giá trị cao nhất có thể .

Scrum thì :

* Gọn nhẹ
* Đơn giản để hiểu
* Khó để thành thục

Scrum là mô hình quy trình được sử dụng để quản lý phát triển sản phẩm phức tạp kể từ đầu những năm 1990 . Scrum không phải là một quy trình hoặc một kỹ thuật để xây dựng sản phẩm. Đúng hơn, nó là một mô hình trong đó bạn có thể sử dụng nhiều quy trình và kỹ thuật khác nhau. Scrum làm rõ ràng về hiệu quả tương đối của thực tiễn phát triển và quản lí sản phẩm của bạn từ đó nâng cao giá trị sản phẩm.

Mô hình Scrum bao gồm Scrum Team và các role liên quan, các event, artifact, và các rule . Mỗi thành phần trong mô hình phục vụ một mục đích cụ thể và cần thiết trong việc áp dụng và sự thành công của Scrum .

### Nguyên lí của Scrum

Scrum được thành lập trên lý thuyết điều khiển quá trình thực nghiệm, hoặc chủ nghĩa kinh nghiệm. Chủ nghĩa kinh nghiệm khẳng định rằng kiến thức xuất phát kinh nghiệm và ra quyết định dựa trên những gì được biết đến. Scrum sử dụng một cách tiếp cận lặp, tăng trưởng để tối ưu hóa khả năng dự đoán và kiểm soát rủi ro. Ba trụ cột duy trì thực hiện kiểm soát quá trình thực nghiệm: tính minh bạch, kiểm tra, và thích ứng.