

1

TỔNG QUAN AGILE

Trong chương này...

- Những vấn đề nổi cộm trong phát triển sản phẩm
- Sự ra đời của Agile
- Tuyên ngôn Phát triển Phần mềm Linh hoạt
- Cách tiếp cận lặp tăng trưởng và truyền thống
- Khi nào Agile, khi nào không?



Các phương pháp Agile tối ưu các tri thức ẩn trong đội ngũ phát triển, thay vì lệ thuộc vào những thông tin được viết ra dưới dạng kế hoạch hay tài liệu.

Barry Boehm

Những vấn đề phổ biến trong phát triển sản phẩm và quản trị dự án phần mềm

Ngành phát triển phần mềm đã có hơn nửa thế kỉ hình thành và phát triển, tuy nhiên việc phát triển phần mềm và quản trị các dự án phần mềm chưa bao giờ hết thử thách. Rất nhiều những vấn đề cứ không ngừng lặp đi lặp lại, làm đau đầu đội ngũ phát triển và các nhà quản trị. Dưới đây là một số vấn đề tiêu biểu.

Theo cách thức phát triển truyền thống, dự án được lập kế hoạch cẩn thận, được tiến hành với rất nhiều khâu trung gian, và khách hàng ngồi chờ. Các bên liên quan hầu như chỉ nhận được báo cáo, tài liệu chứ không nhận được phần mềm để dùng. Thông tin về phẩm mềm rất mù mờ. Lâu tới ngày phát hành và không minh bạch về tiến độ khiến khách hàng và các bên liên quan sốt ruột, lo lắng và không làm cách nào để cung cấp các phản hồi có ý nghĩa, khó hợp tác với đội phát triển cho ra sản phẩm tốt nhất.

Kế hoạch đã định, nhưng nhiều rủi ro xuất hiện: lỗi xuất hiện ngày càng nhiều, hiểu sai yêu cầu, những khó khăn về mặt công nghệ, một vài thành viên có vấn đề và không làm việc như kỳ vọng và *sản phẩm chậm ngày phát hành*.

Mọi thứ đã được tích hợp, nhưng *sản phẩm thiếu ổn định* do không kiểm soát được chất lượng. Ở rất nhiều dự án, mọi công việc đã hoàn thành nhưng giai đoạn tích hợp và ổn định hệ thống thật là thâm họa, không biết khi nào kết thúc.

Dừng và Nghĩ

Bạn và nhóm của mình đang gặp phải những vấn đề nổi cộm nào?

Kế hoạch thường xuyên sai sót nên chúng ta phải đầu tư nhiều thời gian hơn vào xây dựng kế hoạch. Nhưng dù cố gắng đến mấy thì kế hoạch vẫn không đầu vào đầu và chúng ta lại bị phàn nàn là làm kế hoạch chưa kỹ. *Lập kế hoạch thật là một ác mộng*, nhiệm vụ không thể hoàn thành.

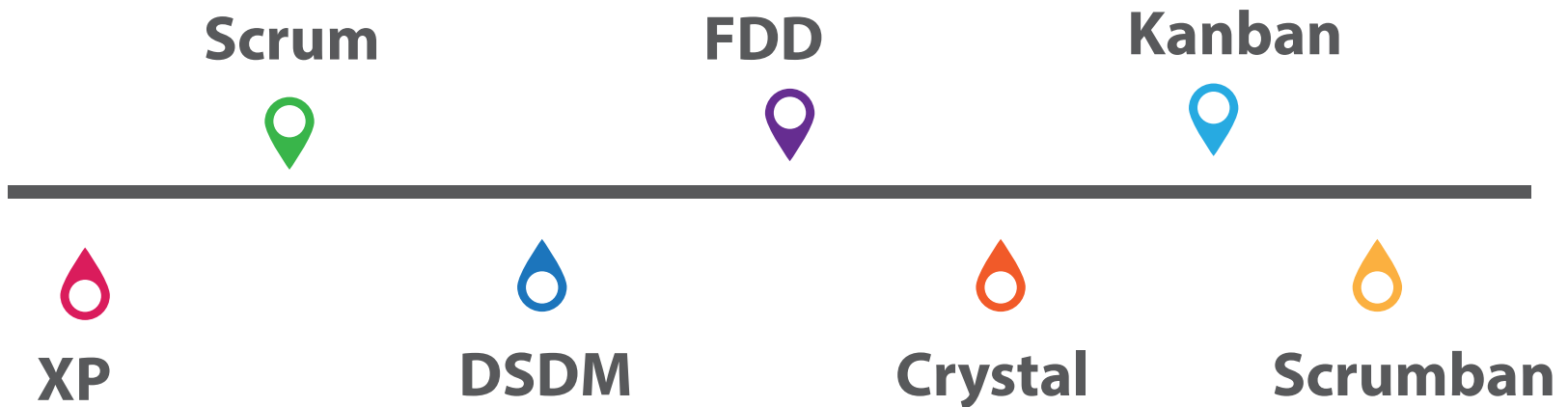
Khi đã thực hiện xong phân tích yêu cầu, thiết kế và bắt đầu lập trình, khách hàng đổi yêu cầu. Vì đó là yêu cầu rất quan trọng, bạn không thể từ chối. Nhóm rất khó xử. Họ không biết làm thế nào bởi *việc thay đổi rất khó*.

Ban đầu sản phẩm còn chạy tốt, nhưng lượng mã nguồn ngày một tăng lên, áp lực thời gian, v.v. nên *chất lượng sản phẩm cứ giảm sút*. Kế hoạch đã bị vỡ, chất lượng ngày một giảm, tất cả mọi người phải làm thêm giờ với áp lực rất cao, thành viên không hạnh phúc.

KHỦNG HOẢNG PHƯƠNG PHÁP LUẬN, VÀ SỰ RA ĐỜI CỦA AGILE

Thập kỉ 80 của thế kỉ XX chứng kiến một thời kì khủng hoảng phương pháp luận phát triển phần mềm, do tỉ lệ thất bại của các dự án phần mềm quá cao. Hàng loạt nỗ lực của các nhà thực hành cũng như giới hàn lâm đã cố gắng tìm ra phương pháp hữu hiệu để đảm bảo tính hiệu quả trong phát triển phần mềm.

Thập kỉ 90, nhiều phương pháp phát triển phần mềm với quy trình nhẹ (light weight) và linh hoạt ra đời, như XP, Scrum, FDD, Crystal, ... và nhanh chóng được lan rộng.



Từ 11 - 13 tháng 2 năm 2001, 17 nhà phát minh và nhà thực hành đã họp 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. Ngày nay chúng ta gọi Agile, tức là chỉ chung một họ phương pháp phát triển phần mềm có chung sự chia sẻ các giá trị và nguyên tắc được phát biểu trong Tuyên ngôn Agile. Biện pháp thực hành có thể rất khác nhau, nhưng triết lý chung thì giống nhau.



*Sự ra đời Tuyên ngôn Agile.
Ảnh: agilemanifesto.org*

Sau một thập kỉ rưỡi ra đời. Agile đã cải thiện đáng kể khả năng thành công của các dự án.

Báo cáo CHAOS của Standish Group năm 2015 cho thấy, các **dự án Agile** có tỉ lệ thành công cao hơn **3 lần** so với dự án truyền thống

Dự án lớn áp dụng Agile có tỉ lệ thành công cao gấp 6 lần

Dự án vừa áp dụng Agile có tỉ lệ thành công cao gấp gần 4 lần

Dự án nhỏ áp dụng Agile cũng cho tỉ lệ thành công cao hơn

Quy mô Dự án	Phương pháp
--------------	-------------

Tổng kết	Agile
	Waterfall

Lớn	Agile
	Waterfall

Vừa	Agile
	Waterfall

Nhỏ	Agile
	Waterfall

Dự án thành công: đúng thời gian, đúng ngân sách với mọi tính năng và kết quả thoả mãn.

Dự án thử thách: hoàn thành nhưng không đạt một trong các tiêu chí đúng ngân sách, đúng thời gian, hoặc kết quả không thoả mãn

Dự án thất bại: dự án bị cắt ở một thời điểm nào đó trong quá trình phát triển

Thành công	Thử thách	Thất bại
39%	52%	9%
11%	60%	29%
18%	59%	23%
3%	55%	42%
27%	62%	11%
7%	68%	25%
58%	38%	4%
44%	45%	11%

Những nghiên cứu chi tiết hơn định lượng được tác động của những phương pháp Agile lên năng suất lao động.

Như trong nghiên cứu của Sutherland và đồng nghiệp năm 2008, năng suất của nhóm Scrum cao hơn tới gần 9 lần so với nhóm sử dụng phương pháp

truyền thống.

Trong nghiên cứu này, kích thước phần mềm được đo bằng hai thước đo phổ biến trong phát triển phần mềm là số dòng mã và điểm chức năng (function point).

Năng suất tính theo điểm chức năng

của nhóm Scrum cao hơn nhóm truyền thống $15,3/1,7 = 8,88$ lần.

Loại nhóm ----- Chỉ số	Nhóm Scrum tập trung	Waterfall	Nhóm Scrum phân tán Sirsi Dynix	Nhóm Scrum phân tán Xebia
Người * tháng	54	540	827	125 (-77%) *
Số dòng mã Java	51.000	58.000	671.668	100.000 (-42%) *
Function points (FP)	959	900	12.673	1.887 (+210%) *
FP cho mỗi KLOC **	18,80	15,52	18,86	18,87 (+22%) *
Lượng FP trên người * tháng	17,8	1,7	15,3	15,1 (+888%) *

* So sánh với phương pháp truyền thống waterfall
** KLOC: 1000 dòng mã

TUYÊN NGÔN PHÁT TRIỂN PHẦN MỀM LINH HOẠT

*Chúng tôi đã tìm ra cách tốt hơn để phát triển phần mềm thông qua việc thực hành
và giúp đỡ người khác thực hiện. Qua đó, chúng tôi đánh giá cao:*

Cá nhân và 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 thay đổi

hơn là bám sát kế hoạch.

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

Theo AgileManifesto.org

12

nguyên lý phía sau tuyên ngôn Agile

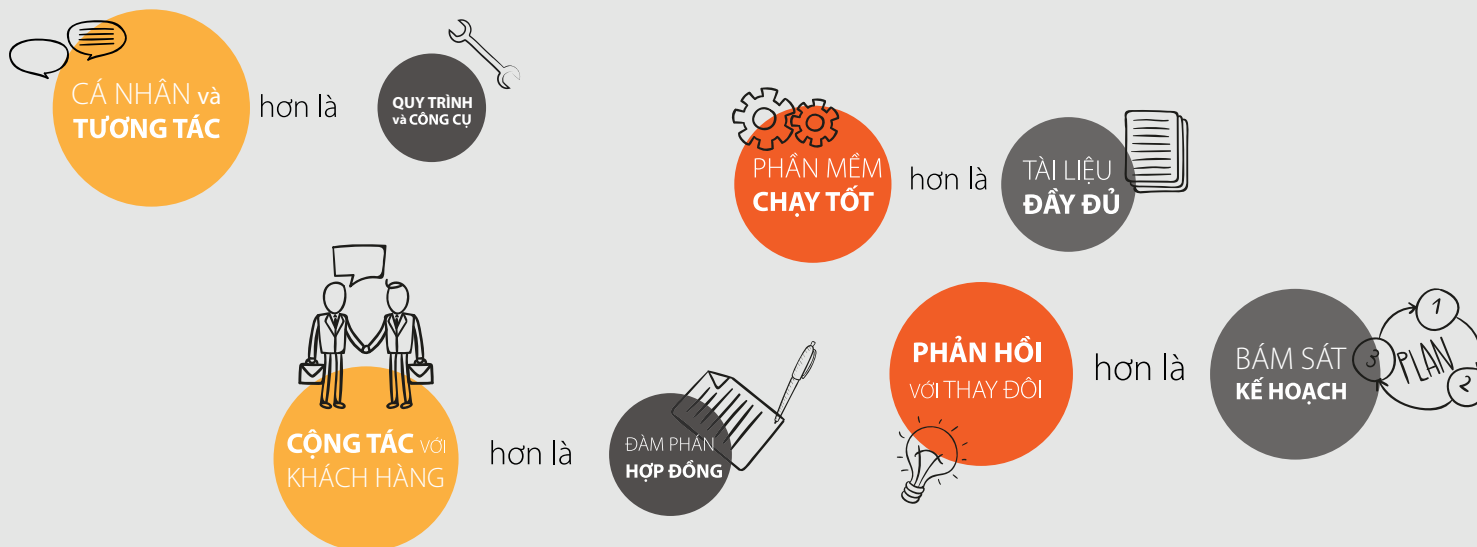
1. Ưu tiên cao nhất của chúng tôi là thỏa mãn khách hàng thông qua việc chuyển giao sớm và liên tục các phần mềm có giá trị.
2. Chào đón việc thay đổi yêu cầu, thậm chí rất muộn trong quá trình phát triển. Các quy trình linh hoạt tận dụng sự thay đổi cho các lợi thế cạnh tranh của khách hàng.
3. Thường 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.
4. Nhà kinh doanh và nhà phát triển phải làm việc cùng nhau hàng ngày trong suốt dự án.
5. Xây dựng các dự án xung quanh những cá nhân có động lực. Cung cấp cho họ môi trường và sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc.
6. Phương pháp hiệu quả nhất để truyền đạt thông tin tới nhóm phát triển và trong nội bộ nhóm phát triển là hội thoại trực tiếp.
7. Phần mềm chạy tốt là thước đo chính của tiến độ.
8. 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.
9. Liên tục quan tâm đến các kĩ thuật và thiết kế tốt để gia tăng sự linh hoạt.
10. Sự đơn giản – nghệ thuật tối đa hóa lượng công việc chưa xong – là căn bản.
11. Các kiến trúc tốt nhất, yêu cầu tốt nhất, và thiết kế tốt nhất sẽ được làm ra bởi các nhóm tự tổ chức.
12. 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.

Theo AgileManifesto.org

Ý NGHĨA CỦA TUYÊN NGÔN AGILE

theo Jeff Sutherland*

Những giá trị đó không chỉ là thứ mà các tác giả của Tuyên ngôn Agile dự định cung cấp ra để phục vụ cho tuyên ngôn và rồi sau đó đi vào quên lãng. Chúng là các giá trị căn cứ vào để làm việc. Bản thân mỗi phương pháp linh hoạt đều tiếp cận các giá trị trên theo các cách thức khác nhau, nhưng tất cả các phương pháp này đều có tiến trình và biện pháp thực hành cụ thể để nuôi dưỡng một hoặc nhiều trong số các giá trị đó.



Cá nhân và tương tác

Cá nhân và sự tương tác giữa họ là cốt yếu để nhóm đạt được hiệu suất cao. Các nghiên cứu về sự “bão hòa giao tiếp” (communication saturation) trong 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 giao tiếp, các phương pháp linh hoạt thường xuyên sử dụng chu trình thanh-tra-và-thích-nghi. 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 meeting (họp đứng), 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 giao tiếp. Chu kỳ thanh-tra-và-thích-nghi hoạt động 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 giao tiếp
- 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 lí 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 việc hình thành chúng. Hầu hết các nhóm tránh né 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 xuất phát từ truyền thống trước đây. Để chống lại những khuynh hướng này, ban lãnh đạo và thành viên nhóm phải tạo điều kiện cho những xung đột tích cực. Làm như vậy không chỉ giúp tạo ra hành vi sinh lợi mà còn đem lại các lợi ích khác:

- 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.
- Đối mới chỉ xảy ra với việc tự do trao đổi các ý tưởng mâu thuẫn nhau, như Takeuchi và Nonaka đã từng nghiên cứu và chỉ ra.
- 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 của 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 và 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 nhóm đưa ra một danh sách công việc được ưu tiên hóa, để họ tự quản lí công việc của mình, và tập trung vào cải tiến về cách thực hiện các công việc đó.



Phần mềm chạy tốt

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 tỉ 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.



hơn là



Cộng tác với khách hàng

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 Agile đã làm sáng tỏ điều này khi họ nhấn mạnh rằng việc 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. 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. 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 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 hằng ngày. Sử dụng 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 hằng 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 dành riêng cho vị khách hàng đại diện này.

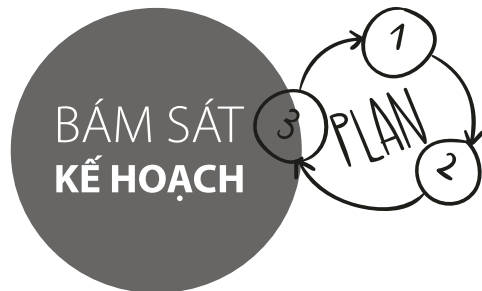
Phản hồi với thay đổi

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, trong giới hạn 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.



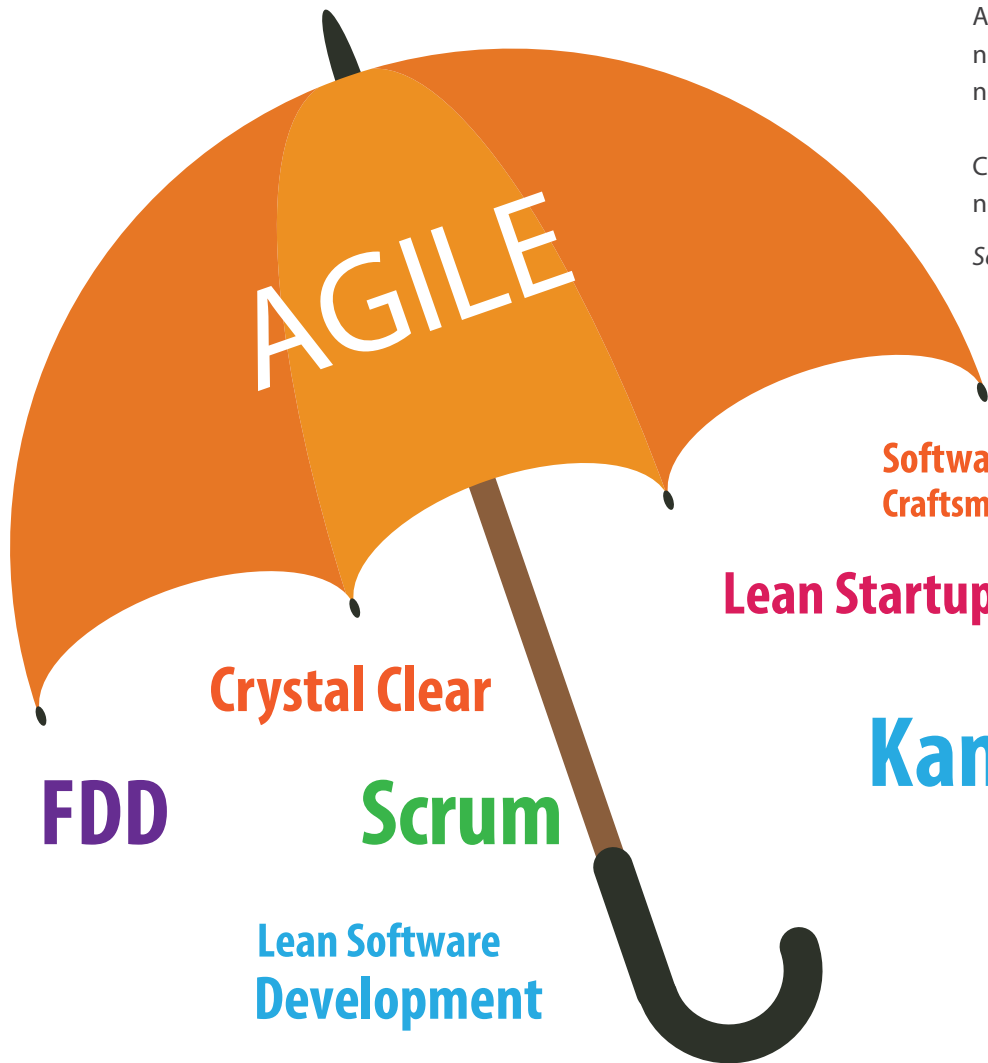
hơn là

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.



Dừng và Nghĩ

Bạn có đồng tình với các điểm chính của Tuyên ngôn Agile không? Bạn có muốn bổ sung gì thêm không?



Agile chỉ là tập hợp những giá trị được nêu trong Tuyên ngôn Phát triển Phần mềm Linh hoạt (gọi tắt là Tuyên ngôn Agile).

Có rất nhiều phương pháp chia sẻ những giá trị của Tuyên ngôn Agile, chúng đều được gọi là phương pháp Agile

Scrum là một trong số các phương pháp Agile phổ biến.

**Software
Craftsmanship**

Lean Startup

Kanban

Crystal Clear

FDD

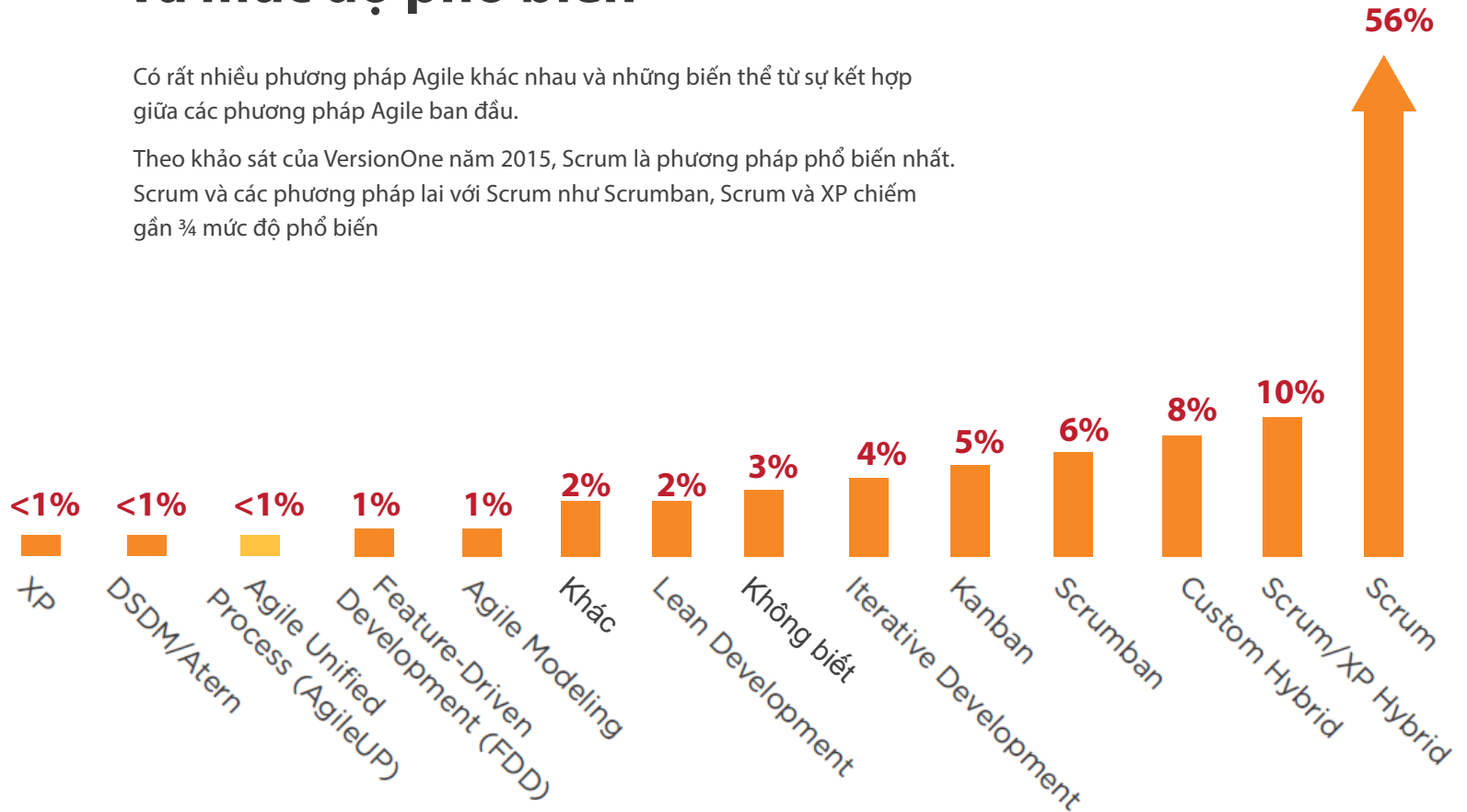
Scrum

**Lean Software
Development**

Các phương pháp Agile và mức độ phổ biến

Có rất nhiều phương pháp Agile khác nhau và những biến thể từ sự kết hợp giữa các phương pháp Agile ban đầu.

Theo khảo sát của VersionOne năm 2015, Scrum là phương pháp phổ biến nhất. Scrum và các phương pháp lai với Scrum như Scrumban, Scrum và XP chiếm gần ¾ mức độ phổ biến

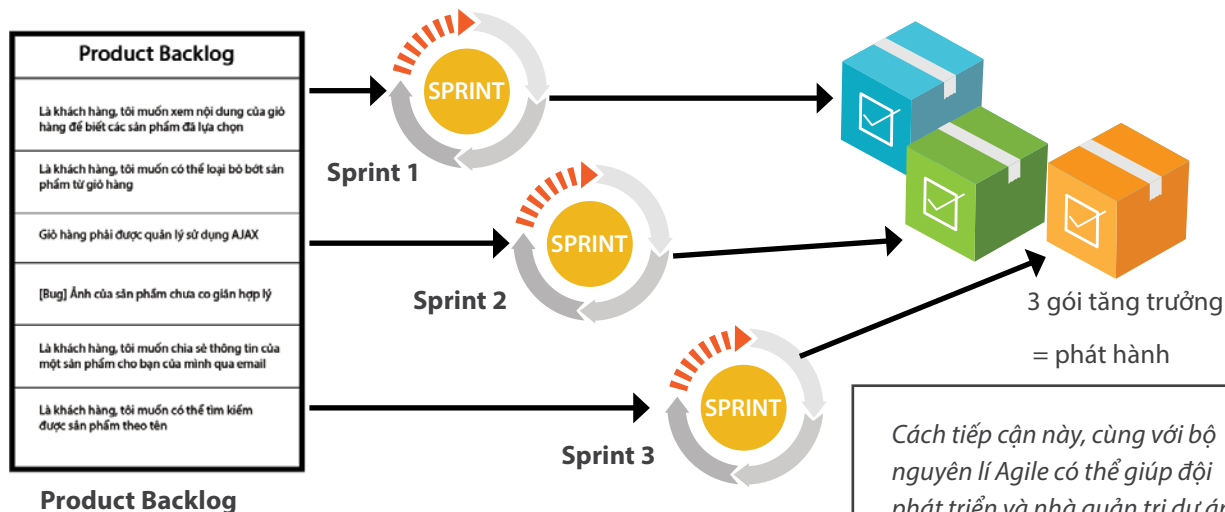


Đặc trưng quan trọng của Agile: Tiếp cận tăng trưởng lặp

Nhóm Phát triển sẽ hoàn thành một tập những tính năng ở phần trên của Product Backlog trong một khoảng thời gian đóng khung (Sprint) tạo ra một phần tăng trưởng cho khách hàng.

Yêu cầu sản phẩm được sắp xếp theo độ ưu tiên. Những hạng mục đem lại nhiều giá trị trên vốn đầu tư (ROI) sẽ ở trên.

Product Backlog chứa các yêu cầu về sản phẩm được cập nhật liên tục để thích ứng với những phản hồi với những thay đổi cần thiết.

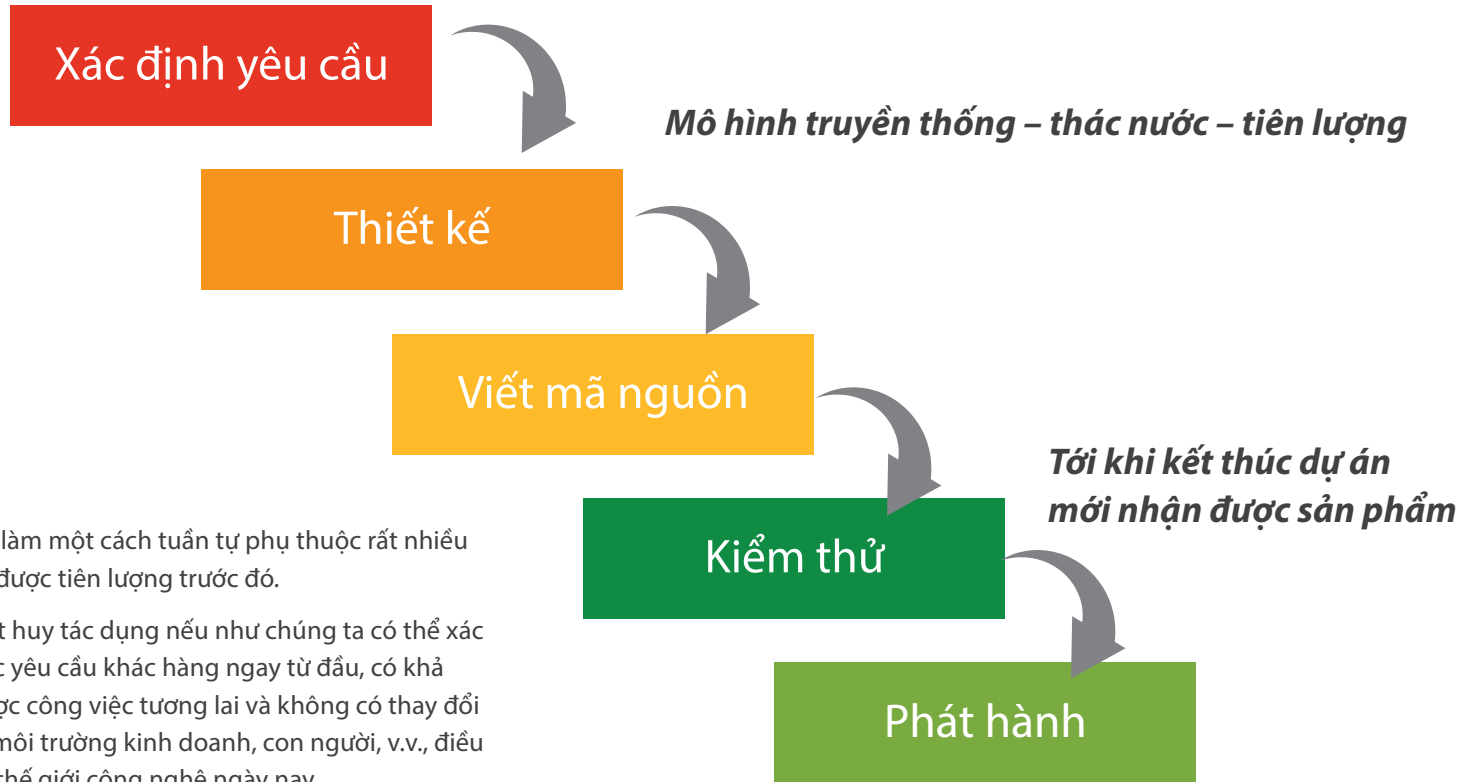


Khách hàng sẽ nhận được phần tăng trưởng cuối mỗi Sprint.

Sản phẩm sẽ tăng trưởng dần cùng với thời gian, tuy nhiên tại bất cứ thời điểm nào họ cũng có phần mềm chạy tốt.

Cách tiếp cận này, cùng với bộ nguyên lý Agile có thể giúp đội phát triển và nhà quản trị dự án vượt qua được hầu hết những vấn đề được nhắc tới trong phần đầu của chương này.

Trước Agile, phần mềm cơ bản được phát triển theo mô hình thác nước (Waterfall), hoặc dựa theo kế hoạch (plan-driven)



Các công việc được làm một cách tuần tự phụ thuộc rất nhiều vào những điều đã được tiên lượng trước đó.

Mô hình này sẽ phát huy tác dụng nếu như chúng ta có thể xác định được chính xác yêu cầu khách hàng ngay từ đầu, có khả năng tiên lượng được công việc tương lai và không có thay đổi nào về công nghệ, môi trường kinh doanh, con người, v.v., điều rất hiếm gặp trong thế giới công nghệ ngày nay.

KHI NÀO AGILE, KHI NÀO KHÔNG

Các dự án có quy mô, đặc thù và độ phức tạp không giống nhau, do vậy chúng ta cần phải có những phương pháp phát triển phù hợp.

Để xác định được khi nào thì ra quyết định kiểu gì, làm việc như thế nào, chúng ta có thể sử dụng Mô hình Cynefin cho việc phân loại tình huống và dự án, rồi ra các quyết định quan trọng.

Theo đó có các vùng chính mô tả các đặc trưng khác nhau bao gồm Hiện nhiên, Rầm rối, Phức hợp, Hỗn độn.

Bảng sau đây liệt kê các đặc tính từng vùng và vai trò của người ra quyết định.



Mô hình Cynefin cho việc ra quyết định chiến lược

Đối với các dự án phát triển sản phẩm mới, thông thường chúng rơi vào vùng Phức hợp. Đó là khu vực mà phương pháp truyền thống sẽ gặp khó khăn, và cách tiếp cận Agile sẽ phù hợp hơn.

Tình huống	Đặc điểm nhận dạng	Vai trò người ra quyết định
Hiển nhiên	<ul style="list-style-type: none"> • Có một câu trả lời đúng đắn • Mọi người biết rõ những cái đã biết • Quản lí theo thực tế (Fact-based management) 	<ul style="list-style-type: none"> • Phán đoán trước, phân loại sau, rồi phản hồi • Sử dụng quy trình đúng đắn • Ủy quyền • Sử dụng kinh nghiệm tốt (best practice) • Truyền đạt rõ ràng, trực tiếp • Giao tiếp hai chiều có thể không cần thiết
Rắc rối	<ul style="list-style-type: none"> • Cần chuyên gia phân tích • Có thể nhận biết quan hệ nhân-quả nhưng không hiển nhiên đối với mọi người • Có nhiều hơn một câu trả lời • Mọi người biết là có những thứ chưa biết đến • Quản lí theo thực tế 	<ul style="list-style-type: none"> • Phán đoán trước, phân tích sau, rồi phản hồi • Thiết lập đội chuyên gia • Lắng nghe các lời khuyên có phần khác biệt
Phức hợp	<ul style="list-style-type: none"> • Thay đổi liên tục và không thể tiên lượng • Không có câu trả lời đúng đắn; các khuôn mẫu hoạt động được ló dần ra • Mọi người không biết những điều chưa biết • Rất nhiều ý tưởng đối nghịch nhau • Cần những tiếp cận sáng tạo • Lãnh đạo dựa-trên-khuôn-mẫu (pattern-based leadership) 	<ul style="list-style-type: none"> • Thử nghiệm trước, phán đoán sau, rồi phản hồi • Tạo môi trường thử nghiệm cho phép các khuôn mẫu xuất hiện. • Tăng mức độ tương tác và giao tiếp hai chiều • Dùng các phương pháp để tạo lập thật nhiều ý tưởng: thảo luận mở; tạo lập biên; khuyến khích các nhân tố thu hút (attractor); khuyến khích sự bất đồng quan điểm và sự đa dạng; quản lí các điều kiện ban đầu và giám sát để hình thành các khuôn mẫu.
Hỗn độn	<ul style="list-style-type: none"> • Nhiều loạn cao độ • Không rõ quan hệ nhân- quả, thậm chí không có điểm nào để tìm kiếm manh mối • Không thể biết được điều gì • Phải ra nhiều quyết định cấp tập mà không có thời gian để suy nghĩ • Lãnh đạo dựa-trên-khuôn-mẫu (pattern-based leadership) 	<ul style="list-style-type: none"> • Hành động trước, phán đoán sau, rồi phản hồi • Quan sát xem cái nào hiệu quả thay vì tìm kiếm câu hỏi đúng đắn • Hành động tập lự để văn hồi trật tự (mệnh lệnh và kiểm soát) • Truyền đạt rõ ràng, trực tiếp

NHỮNG CÂU HỎI THƯỜNG GẶP

Agile là gì?

Phát triển Linh hoạt (Agile Development, gọi tắt là Agile) là một thuật ngữ có nguồn gốc từ Tuyên Ngôn Phát triển Phần mềm Linh hoạt (The Manifesto for Agile Software Development). Tuyên ngôn này được soạn thảo năm 2001 bởi một nhóm các tác giả, chuyên gia của các phương pháp phát triển phần mềm Scrum, Extreme Programming (XP), DSDM (Dynamic Systems Development Method), Crystal, FDD (Feature-Driven Development), và một vài nhà lãnh đạo khác.

Tuyên ngôn Agile đã tổng kết ra một số giá trị và nguyên tắc chung, phổ quát cho tất cả các phương pháp luận về linh hoạt đang tồn tại độc lập tại thời điểm đó.

Khi dùng chữ “agile” với tư cách là một tính từ, thì nó mang nghĩa là “linh hoạt”. Nhưng khi ta dùng chữ Agile (A viết hoa), tức là chỉ triết lí và tất cả những phương pháp phát triển phần mềm, phát triển sản phẩm và quản lí dựa trên triết lí được mô tả trong Tuyên ngôn Agile.

Agile là triết lí đã làm thay đổi diện mạo nền công nghệ thế giới trong những thập kỉ đầu thế kỉ XXI. Các phương pháp Agile đã được thực tế chứng minh là công cụ mạnh của các cá nhân, tổ chức trong việc đổi mới, sáng tạo, làm nên những sản phẩm chất lượng cao, đột phá, cũng như gia tăng hiệu quả hoạt động và năng lực cạnh tranh, phát triển văn hóa tổ chức. Những nghiên cứu thực nghiệm và thống kê từ ngành công nghiệp phần mềm cho thấy Agile có thể giúp cho các doanh nghiệp gia tăng năng suất và hiệu quả lên gấp nhiều lần, nâng tỉ lệ thành công của dự án lên gấp đôi và gia tăng năng lực cạnh tranh bền vững cho các doanh nghiệp. Cùng với Tư duy Tinh gọn (Lean Thinking), Khởi nghiệp Tinh gọn (Lean Startup), và Tư duy Thiết kế (Design Thinking), Agile vẫn đang tiếp tục tạo nên những làn sóng đổi thay lớn trên thế giới.

Scrum có phải là Agile không?

Scrum là một phương pháp Agile (phổ biến nhất) nhưng không phải là Agile. Agile định nghĩa các giá trị cốt lõi và nguyên tắc định hướng, còn Scrum là một phương pháp cụ thể chia sẻ các nguyên tắc đó.

Vậy Scrum là gì?

Tài liệu Hướng dẫn Scrum định nghĩa “Scrum là một khung làm việc để phát triển bền vững các sản phẩm phức tạp”. Có thể hiểu đây là khung tổ chức công việc tổng quát hướng đến phát triển các sản phẩm phức tạp, chủ yếu là phần mềm. Ngày nay Scrum có thể được dùng như là nền tảng tổ chức các công việc khác nhau, từ quản trị dự án linh hoạt nói chung, đến phát triển sản phẩm, thực hiện các chiến dịch marketing, tổ chức dạy học, sản xuất ô tô module hóa hoặc những công việc cá nhân khác.

Dựa trên lý thuyết quản lý thực nghiệm (empirical process control), Scrum sử dụng cơ chế lặp (iterative) và tăng trưởng (incremental) để tối ưu hóa tính hiệu quả và kiểm soát rủi ro. Scrum rất đơn giản, dễ học và có khả năng ứng dụng rất rộng. Để có thể dùng Scrum, chúng ta cần hiểu rõ và vận dụng đúng các thành tố tạo nên Scrum bao gồm các giá trị cốt lõi (còn gọi là “ba chân”, hay ba trụ cột của Scrum), các vai trò, các sự kiện, các tạo tác (artifact) và những quy tắc đặc thù của Scrum.

Hiện nay, định nghĩa Scrum là gì được ghi trong tài liệu Scrum Guide (Hướng dẫn Scrum) và được cập nhật thường xuyên bởi chính các tác giả tại <http://scrumguides.org>.

Toàn bộ cuốn sách này sẽ là về Scrum. Hãy đọc tiếp các chương sau.

Tại sao chọn Agile?

Trong nền kinh tế sáng tạo đầy biến động hiện nay, yêu cầu cấp thiết đối với các cá nhân hay tổ chức là phải đạt được năng lực linh hoạt (Agility) cao độ để thích ứng và dẫn đầu trong các cuộc cạnh tranh. Agile là nền tảng để đạt được Agility và duy trì lợi thế cạnh tranh bền vững.

Triết lí Agile cùng với các framework, phương pháp cũng như các công cụ đặc thù có thể giúp các cá nhân, đội nhóm và doanh nghiệp trở nên hiệu quả hơn, năng suất hơn và đổi mới-sáng tạo tốt hơn.

Muốn triển khai Agile thì bắt đầu từ đâu?

Các cá nhân có thể bắt đầu với việc áp dụng những kĩ thuật cơ bản của Personal Kanban cùng với tư duy Agile để nâng cao năng suất cá nhân, làm việc hiệu quả, kiểm soát tốt stress.

Các lập trình viên có thể bắt tay với những kĩ thuật lập trình Agile cơ bản như TDD, Pair-Programming để nâng cao chất lượng phần mềm trước khi đi xa hơn với đầy đủ bộ kĩ năng để trở thành Nghệ nhân Phần mềm (Software Craftsman) đích thực.

Các nhóm có thể ứng dụng Scrum để tổ chức lại công việc, nâng cao hiệu quả giao tiếp và hiệu suất nhóm.

Các công ty có thể bắt đầu với việc tái tổ chức công việc và quy trình nghiệp vụ với Scrum, Lean để giảm thiểu lãng phí, nâng cao hiệu quả hoạt động, tối ưu hóa nguồn lực và gia tăng năng lực cạnh tranh bền vững.

Triển khai Agile có tốn kém không? Có đòi hỏi điều kiện gì đặc biệt về trình độ không?

Phụ thuộc vào nguồn lực chúng ta có, việc thực hành Agile có thể đơn giản là sắp xếp lại công việc mà không cần thêm chi phí. Đối với các chương trình chuyển đổi căn bản và toàn diện, sự đầu tư có thể sẽ phải tương xứng với mục tiêu do tổ chức đề ra.

Có bộ công cụ hay phần mềm nào để thực hành Agile?

Agile quy trình tư duy, trong khi phương pháp sẽ định hình cách triển khai cụ thể, và các công cụ sẽ trợ giúp một phần nào đó. Công cụ và quy trình cần được lựa chọn kĩ để phục vụ các cá nhân làm việc hiệu quả và tương tác nhóm tốt hơn, không nên là lựa chọn để “đánh dấu” rằng “tôi đã sử dụng Agile”.

Các nhóm Scrum nhỏ, làm việc tập trung có thể chỉ cần các đồ chi phí thấp để có thể vận hành tốt Scrum: bảng trắng, giấy dán, bút viết bảng.

Chúng ta cũng có thể sử dụng hàng loạt các hệ thống phần mềm, cloud hỗ trợ tương tác nhóm theo triết lí Agile như Trello, Asana, Jira, TFS. Chúng có thể miễn phí hoặc tính phí tùy nhu cầu sử dụng của mỗi nhóm và tổ chức.

Đối với các tổ chức lớn, việc đầu tư cho các công cụ để gia tăng hiệu quả tổng thể luôn là một trong những ưu tiên trong danh mục đầu tư cho công nghệ để phát triển.



CÂU HỎI ỨNG DỤNG

1. Tuyên ngôn Agile giúp bạn nhận ra giá trị gì cho mình?
2. Có người diễn dịch ý thứ hai trong Tuyên ngôn Agile là “phần mềm chạy tốt thay vì tài liệu đầy đủ”, theo bạn diễn dịch như vậy có gì không đúng? Tại sao?
3. Nhiều người cho rằng Tuyên ngôn Agile có 5 điều, theo bạn điều thứ 5 của bản tuyên ngôn này là gì?
4. Căn cứ vào đâu để nói rằng một phương pháp phát triển phần mềm là Agile?
5. Bạn nhận định xem những lý do nào khiến các phương pháp phát triển phần mềm truyền thống không còn phù hợp với thực tế?
6. Phân loại dự án của bạn theo mô hình Cynefin?
7. Tại sao cách thức phát triển lặp (iterative) và tăng trưởng (incremental) lại giúp các phương pháp Agile thích ứng với những thay đổi từ khách hàng, môi trường kinh doanh, công nghệ, v.v.?
8. Bạn sẽ giải thích như thế nào với đồng nghiệp cho rằng làm Agile là không cần có kế hoạch?
9. Khi thực hiện một dự án cụ thể của bạn thì cần tạo bao nhiêu tài liệu là đủ?
10. Hãy phân tích những yếu tố công cụ, quy trình nào đang cản trở động lực làm việc của cá nhân và sự cộng tác trong nhóm/tổ chức của bạn?