

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 1/16

MỤC LỤC

A.	MỤC ĐÍCH	2
B.	PHẠM VI	2
C.	THAM KHẢO	2
D.	THUẬT NGỮ	2
1.	AGILE	2
1.1	Agile là gì?	2
1.2	Các phương pháp Agile	2
1.3	Tuyên ngôn phát triển phần mềm linh hoạt	3
1.4	Nguyên lý	3
1.5	Agile Mindset	4
2.	SCRUM	4
2.1	Định nghĩa	5
2.2	Giá trị cốt lõi	6
2.3	Tính cách cần có với thành viên trong Scrum	6
E.	NỘI DUNG	7
I.	TỔ CHỨC QUYỀN VÀ NGHĨA VỤ TRONG SCRUM	7
1.	Product Owner (PO)	7
2.	ScrumMaster	7
3.	Nhóm phát triển	8
4.	Nhóm hỗ trợ khác	9
II.	CÁC BƯỚC PHÁT TRIỂN DỰ ÁN THEO SCRUM	11
1.	Lập kế hoạch Sprint	11
2.	Scrum hàng ngày (daily meeting)	12
3.	Sơ kết Sprint	13
4.	Cải tiến Sprint	13
III.	LƯU Ý KHÁC	13
1.	Version	13
2.	Product Backlog	14
3.	Làm mịn Product Backlog	15
4.	User Stories	15
5.	Định nghĩa hoàn thành	16
6.	Meeting minutes	16
IV.	HỒ SƠ	16

	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 2/16

A. MỤC ĐÍCH

Quy trình phát triển phần mềm nhằm đưa ra các cách thức để phát triển một dự án. Làm rõ vai trò và trách nhiệm của mỗi thành viên trong nhóm phát triển dự án.

B. PHẠM VI

Áp dụng cho các dự án phần mềm theo mô hình làm việc:

- Fixed cost
- Team bases

C. THAM KHẢO

Các tài liệu liên quan

Quy định sử dụng Redmine	ADM-HR-004
Test cases	
Meeting minutes	

D. THUẬT NGỮ

1. AGILE

1.1 Agile là gì?

Agile là một **tập hợp các nguyên lý** dành cho phát triển phần mềm, trong đó khuyến khích việc lập kế hoạch thích ứng, phát triển tăng dần, chuyển giao sớm, và cải tiến liên tục. Agile **không định nghĩa một phương pháp cụ thể** để đạt được những điều này, nhưng lại có nhiều phương pháp phát triển phần mềm khác nhau thỏa mãn và hướng theo các tiêu chí đó.

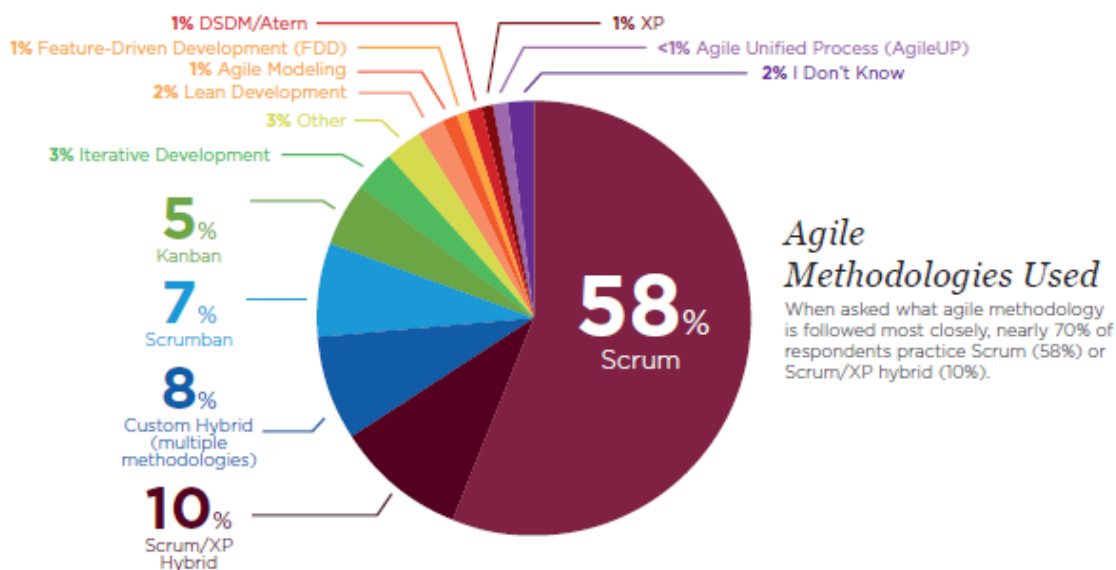


1.2 Các phương pháp Agile

Sau đây là danh sách các phương pháp Agile phổ biến:

- Kanban
- Scrum
- ASD – Adaptive Software Development
- Agile Modeling...

AGILE METHODS AND PRACTICES



1.3 Tuyên ngôn phát triển phần mềm linh hoạt

Cá nhân 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 và thay đổi	Hơn là	Bám sát kế hoạch

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

1.4 Nguyên lý

- 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 của 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.
- 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 họ môi trường và sự hỗ trợ cần thiết, và tin tưởng họ để hoàn thành công việc.

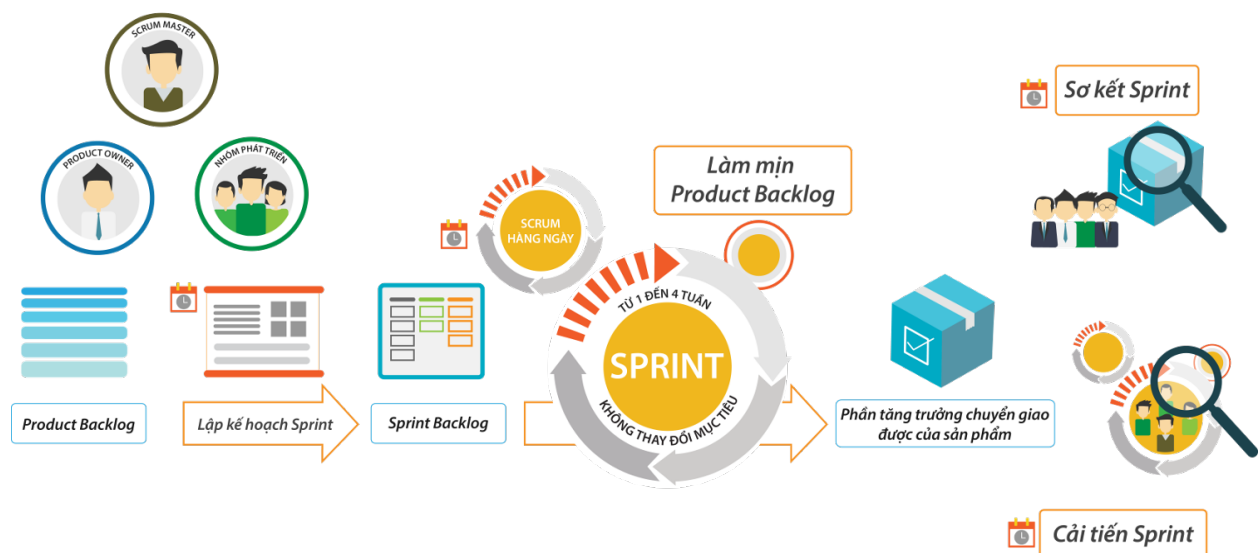
- Phương pháp hiệu quả nhất để truyền đạt thông tin tới nhóm phát triển và trong nội bộ nhóm phát triển là hội thoại trực tiếp.
- Phần mềm chạy tốt là thước đo chính của tiến độ.
- Các quy trình linh hoạt thúc đẩy phát triển bền vững. Nhà đầu tư, nhà phát triển và người dùng có thể duy trì một nhịp độ 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.

1.5 Agile Mindset


- Mục tiêu công việc là làm cho khách hàng thoải mái (delight clients).
- Giảm nhỏ phạm vi công việc (downscaling works), trao quyền nhiều hơn, gia tăng tính tự trị cho nhóm nhỏ và cá nhân.
- Dùng dữ liệu thực tiễn (evidence-based – tức dựa trên bằng chứng) từ các chu trình phản hồi ngắn (short-feedback cycle) để ra quyết định và liên tục cải tiến.
- Luôn duy trì khả năng thích nghi.

2. SCRUM

Scrum là một phương pháp Agile dùng cho phát triển sản phẩm, đặc biệt là phát triển phần mềm. Scrum là một khung quản lý dự án được áp dụng rất rộng rãi, từ những dự án đơn giản với một nhóm phát triển nhỏ cho đến những dự án có yêu cầu rất phức tạp với hàng trăm người tham gia, và kể cả những dự án đòi hỏi khung thời gian cố định.



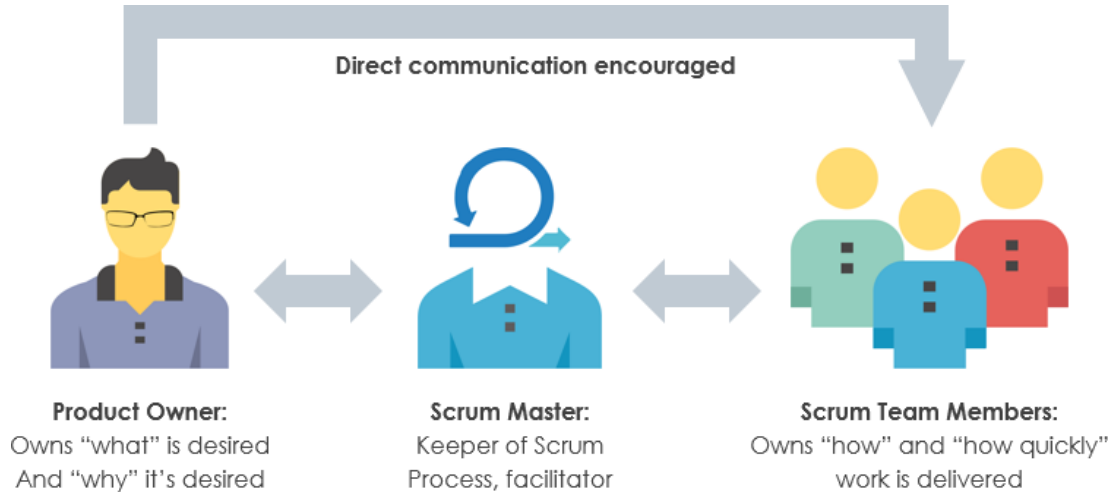
Lưu ý: Scrum là một khung làm việc (framework) chứ không phải một phương pháp (method) cụ thể. Khi vận dụng, Scrum cung cấp các nền tảng cơ bản, kết hợp với các phương pháp hay biện pháp thực hành khác để phát huy tác dụng.

	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 5/16

2.1 Định nghĩa

a. Vai trò

Là một cá nhân, hoặc một nhóm tham gia vào trong quá trình vận hành của Scrum. Trong Scrum có 3 vai trò chính: Product Owner, ScrumMaster và Nhóm Phát triển.

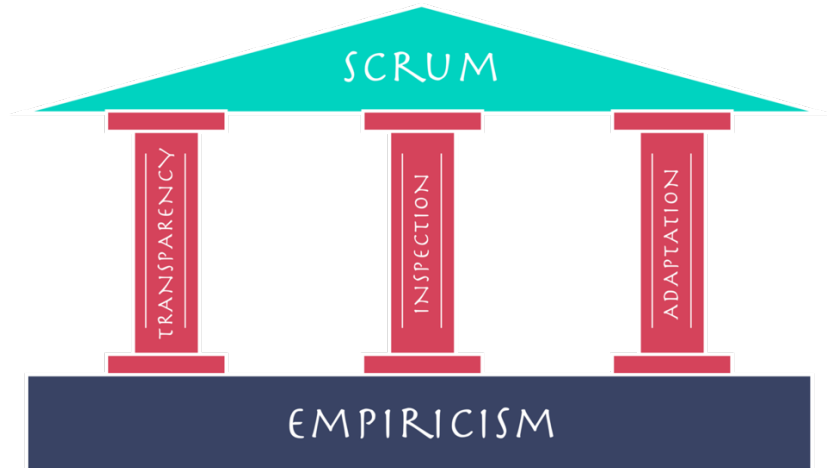


- **Product Owner:** Là người định nghĩa các yêu cầu và đánh giá cuối cùng đầu ra của các nhà phát triển phần mềm.
- **Scrum Master:** Là người có hiểu biết sâu sắc về Scrum và đảm bảo nhóm có thể làm việc hiệu quả với Scrum. Là người chịu trách nhiệm về sự thành công của dự án.
- **Development Team (Nhóm phát triển)** Một nhóm liên chức năng (cross-functional) tự quản lý để tiến hành chuyển đổi các yêu cầu được tổ chức trong Product Backlog thành chức năng của hệ thống.
- b. Sự kiện:** Là các hoạt động xảy ra trong suốt vòng đời của quá trình phát triển. Các sự kiện chính trong Scrum là: Sprint, Lập kế hoạch Sprint, Scrum Hằng ngày, Sơ kết Sprint và Cải tiến Sprint.
- c. Tạo tác :** Là các công cụ hoặc kết quả được tạo ra và sử dụng trong quá trình vận hành Scrum. Các tạo tác trong Scrum bao gồm:
 - **Product Backlog:** Là danh sách các hạng mục cần phát triển của sản phẩm. Danh sách này được sắp xếp theo độ ưu tiên do Product Owner quyết định. Product Owner là người chịu trách nhiệm quản lý Product Backlog, kể cả nội dung, đánh giá độ ưu tiên cho đến việc cập nhật. Các hạng mục trong Product Backlog có thể là các tính năng mong muốn của sản phẩm, hoặc các cải tiến kỹ thuật lớn, hoặc các lỗi kỹ thuật đã được phát hiện...
 - **Sprint Backlog:** Là danh sách chứa các hạng mục Product Backlog được lựa chọn cho Sprint hiện tại và các công việc cần thực hiện để hoàn thành các hạng mục đó. Sprint Backlog có thể chứa thêm thông tin về các ước tính khối lượng công việc của từng hạng mục và của cả Sprint. Sprint Backlog được quản lý bởi Nhóm Phát triển.
 - **Phần tăng trưởng Sản phẩm:** Đôi khi còn được gọi ngắn gọn là Phần tăng trưởng. Là kết quả "hoàn thành" được Nhóm Phát triển chuyển giao sau mỗi Sprint. Để được coi là "hoàn thành" thì các hạng mục Product Backlog phải thỏa mãn với Định nghĩa Hoàn thành đã được thống nhất trước đó. Tính chất cốt lõi của Phần tăng trưởng đó là có khả năng chuyển giao được, tức là có thể đưa vào sử dụng ngay mà không cần làm thêm bất cứ công việc nào nữa.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 6/16

2.2 Giá trị cốt lõi

Scrum là một phương pháp linh hoạt, vì thế nó tuân thủ các nguyên tắc của Tuyên ngôn Agile (Manifesto for Agile Software Development). Ngoài ra Scrum hoạt động dựa trên ba giá trị cốt lõi, còn gọi là Ba chân của Scrum bao gồm: Minh bạch, Thanh tra và Thích nghi.



- **Minh bạch (transparency):** Tính minh bạch được đề cao như là giá trị cốt lõi cơ bản nhất. Muốn thành công với Scrum, thông tin liên quan tới quá trình phát triển phải minh bạch và thông suốt. Các thông tin đó có thể là: tầm nhìn (vision) về sản phẩm, yêu cầu khách hàng, tiến độ công việc, các khúc mắc và rào cản v.v. Từ đó mọi người ở các vai trò khác nhau có đủ thông tin cần thiết để tiến hành các quyết định có giá trị để 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.
- **Thanh tra (inspection):** Công tác thanh tra liên tục các hoạt động trong Scrum 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 dự án. Truy xét kĩ càng và liên tục là cơ chế khởi đầu cho việc thích nghi và các cải tiến liên tục trong Scrum.
- **Thích nghi (adaptation):** Scrum rất linh hoạt như các phương pháp phát triển linh hoạt (agile software development) khác. Nhờ đó nó mang lại tính thích nghi rất cao. Dựa trên các thông tin minh bạch hóa từ các quá trình thanh tra và làm việc, Scrum có thể phản hồi lại các thay đổi một cách tích cực, nhờ đó mang lại thành công cho dự án.

2.3 Tính cách cần có với thành viên trong Scrum

- **Dũng cảm (courage):** Luôn nói ra vấn đề của mình và chấp nhận rất nhiều loại rủi ro khi thay đổi, cam kết. Về cơ bản những giá trị khác không thể có nếu không có sự dũng cảm.
- **Tập chung (focus):** Mọi người tập chung vào công việc trong Sprint & Mục tiêu Sprint của nhóm. Khi nhóm phát triển đã cam kết với những việc trong Sprint, họ cần phải tập chung để hoàn thành những gì đã cam kết.
- **Cam kết (commitment):** Mỗi thành viên cam kết với các thành viên khác về điều mình làm và công việc đã được chọn ở buổi lập kế hoạch team và họp daily.
- **Tôn trọng (respect):** Tôn trọng mọi người trong team, khi không có sự tôn trọng thì mọi người khó thành thật trong chia sẻ.
- **Cởi mở (openness):** Mọi thứ cần rõ ràng, minh bạch để mọi người có thể làm việc hiệu quả.



E. NỘI DUNG

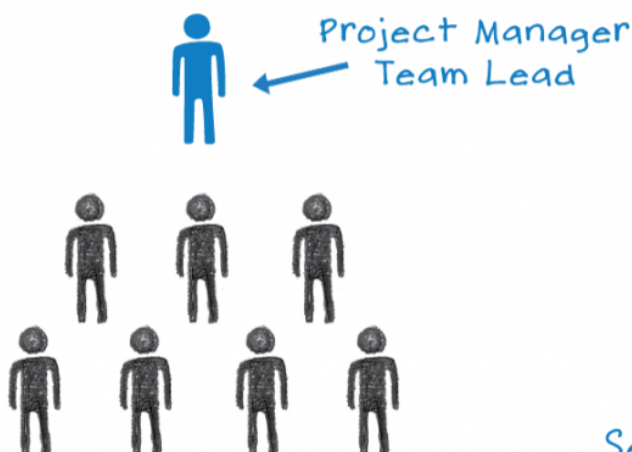
I. TỔ CHỨC QUYỀN VÀ NGHĨA VỤ TRONG SCRUM

1. Product Owner (PO)

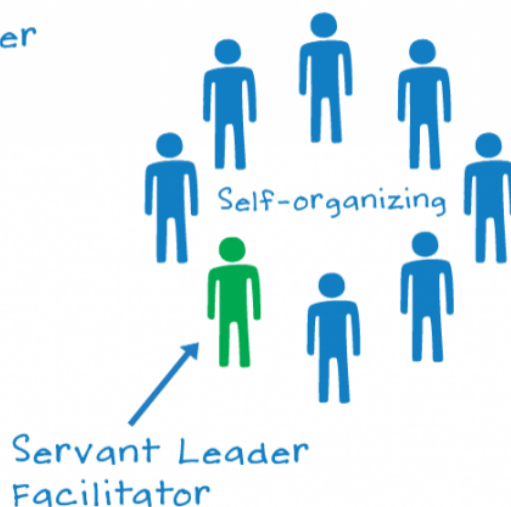
- Phải có tầm nhìn về product và truyền đạt được tầm nhìn đó cho team.
- Người trả lời những câu hỏi liên quan đến requirement của product.
- Quản lý product backlog.
- Tập trung vào WHAT hơn là HOW.
- Chỉ có 1 người duy nhất làm PO.
- Không nhất thiết phải tham gia tất cả các cuộc họp, tuy nhiên cần tham gia các cuộc họp start sprint, start dự án, tổng kết dự án.

2. ScrumMaster

Traditional Teams



Agile Teams



a. Trách nhiệm chung

- Là người lãnh đạo về quy trình chứ không phải là lãnh đạo nhóm.
- Tổ chức các cuộc họp.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 8/16

- Thanh tra, thu thập và minh bạch hóa thông tin. EX: Q/A khách hàng, Trao đổi với PO.
- Loại bỏ trở ngại. Tìm kiếm cải tiến.
- Huấn luyện Scrum cho nhóm.
- Nên là người có hiểu biết kỹ thuật tốt, kinh nghiệm nhiều mảng dự án khác nhau.
- Bám sát tool quản lý Redmine cho mỗi sprint. Kiểm tra tasks của team thường xuyên.
- Trong một số trường hợp ScrumMaster sẽ code một số module nào đó trong dự án.
- Lập kế hoạch dọn Sprint trước đó nếu phát sinh quá nhiều vấn đề.
- Chịu trách nhiệm chính về chất lượng sản phẩm và Kế hoạch của dự án.
- Chịu trách nhiệm về sự thành công của dự án.
- Chuẩn bị yêu cầu của sprint tiếp theo trước khi bắt đầu một sprint mới.
- Luôn tìm mọi giải pháp để đảm bảo đúng tiến độ của sprint mặc dù có thể bị thiếu nguồn lực.

b. Trách nhiệm đối với PO

- Tìm kiếm các kỹ thuật để quản lý hiệu quả project backlog.
- Chia sprint hợp lý và trao đổi với PO khi cần thiết.
- Hiểu rõ việc lập kế hoạch dài hạn sản phẩm trong một môi trường thực nghiệm.
- Hiểu rõ và thực hành sự linh hoạt.
- Thúc đẩy các sự kiện Scrum theo yêu cầu hoặc khi cần thiết.
- Luôn phải trao đổi với PO về vấn đề của team.

c. Trách nhiệm đối với nhóm phát triển.


- Giao tiếp tích cực với nhóm phát triển về tầm nhìn, mục đích và các item của project backlog.
- Huấn luyện team cách tự tổ chức và làm việc liên chức năng.
- Dẫn dắt & điều hướng nhóm phát triển tạo ra các sản phẩm có giá trị cao.
- Loại bỏ các trở ngại trong quá trình phát triển của nhóm.
- Luôn theo dõi sát sao nhóm thông qua Redmine, nhắc nhở nhóm, phản biện dev trong trường hợp đưa ra những estimate vô lý.
- Huấn luyện nhóm hoặc cá nhân trong những trường hợp chưa có kiến thức về Scrum.

d. Một số chú ý cho ScrumMaster khi họp hàng ngày

- Không để daily meeting trở thành buổi báo cáo công việc, hoặc cập nhật tiến độ công việc
- Không để daily meeting biến thành một phiên thảo luận để giải quyết vấn đề.
- Nhóm phát triển không thấy giá trị của buổi Daily meeting.
- Các thành viên thường báo cáo là không gặp vấn đề gì.
- Daily meeting không diễn ra như một thói quen.
- ScrumMaster giao việc cho thành viên trong Daily Meeting.
- Luôn có agenda của meeting, tránh lan man.
- Không quá 15p.
- Phải là người đưa ra quyết định sau cùng cho mỗi cuộc họp.

3. Nhóm phát triển

- Bắt buộc tuân thủ theo quy trình Scrum.
- Luôn cập nhật trạng thái làm việc trên Redmine và tuân thủ theo project backlog.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 9/16

- Phải hiểu rõ tasks của mình làm trước khi nhận tasks. Trong trường hợp không hiểu có thể trao đổi lại với team hoặc yêu cầu cuộc họp kỹ thuật.
- Luôn phải estimate được công việc của mình trong một sprint ngắn (1-2 tuần).
- Khi bắt đầu một task bắt buộc phải đủ 5 yếu tố:
 - Estimate time: ước lượng công việc đủ cho một task, không quá 3 giờ làm việc, nếu quá thì yêu cầu chia nhỏ công việc và trao đổi với team.
 - Start date: Ngày bắt đầu công việc.
 - Due date: Ngày kết thúc công việc.
 - Status: Đổi trạng thái sang in-process.
 - Time logger: Bấm start time khi bắt đầu làm. Lưu ý: Có thể bấm start time khi công việc đó có trạng thái là NEW. Hệ thống sẽ tự chuyển sang trạng thái in-process khi bấm start time.
- Khi kết thúc một task bắt buộc phải các yếu tố dưới đây:
 - Stop time logger: Khi bấm stop time logger hệ thống sẽ tự nhảy đến trang nhập spent time.
 - Trường hợp không có time logger: Yêu cầu nhập ước lượng thời gian mình tốn cho task này. Trường hợp nhập không đúng thì yêu cầu giải trình với team.
 - Status: Đổi trạng thái sang resolved.
 - Description: Nên có mô tả cụ thể công việc mình đã làm. EX: ảnh chụp màn hình, dẫn link
- Khi chuyển sang công việc khác và dừng lại công việc hiện tại.
 - Stop time logger: Khi bấm stop time logger hệ thống sẽ tự nhảy đến trang nhập spent time.
 - Trường hợp không có time logger: Yêu cầu nhập ước lượng thời gian mình tốn cho task này. Trường hợp nhập không đúng thì yêu cầu giải trình với team.
 - Status: Đổi trạng thái sang pending.
 - Description: Ghi rõ lý do tại sao dừng công việc.
- Báo cáo công việc cuối ngày cho QC/Tester. Dọn sạch Redmine trước khi báo cáo.
- Cuối ngày chuẩn bị các công việc tiếp theo trước cuộc họp Daily meeting.
- KHÔNG chat riêng với các members khác về vấn đề công việc của dự án.

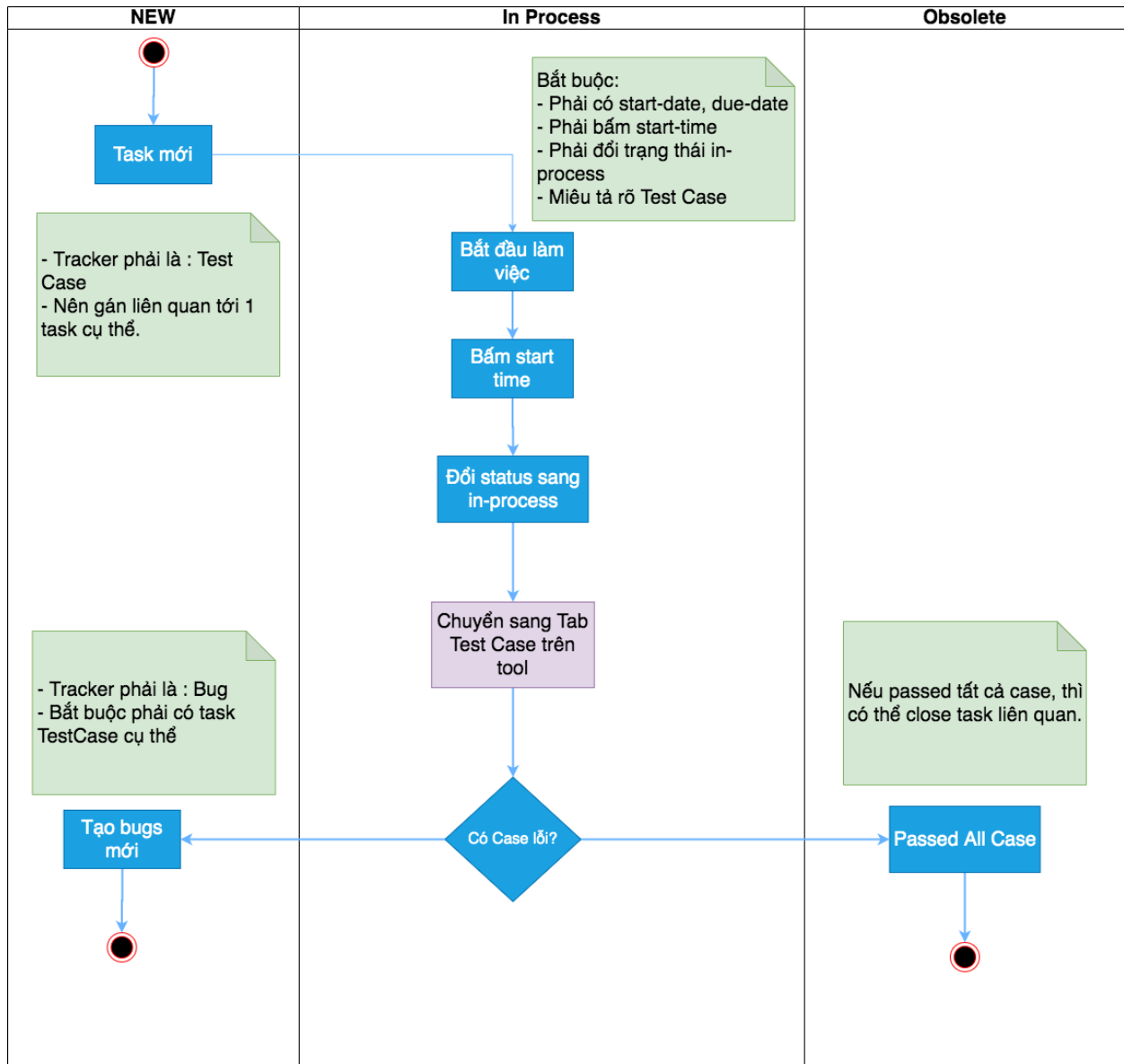
4. Nhóm hỗ trợ khác

Tất cả các thành viên tham gia dự án đều được hiểu là Nhóm phát triển. Tuy nhiên làm rõ vai trò của một số vị trí đặc thù trong nhóm như bên dưới:

a. QC/Tester

- Chịu trách nhiệm về chất lượng dự án.
- Đôn đốc, nhắc nhở dev về workflow làm việc của task. VD: trạng thái task, timelogger, báo cáo..
- Báo cáo tổng kết cuối ngày của dự án.
- Meeting note trong tất cả các cuộc họp của team, cập nhật các tasks liên quan trong cuộc họp.
- Cài đặt các phần mềm tracking theo yêu cầu của team. VD: Telegram để nhận thông tin update tasks của dự án.
- Khi dev hoàn thành một task:
 - Bắt buộc phải có một task TestCase kèm với task của dev.
 - Bắt buộc phải có task TestCase đi kèm khi close một task. Trường hợp không thể có TestCase thì yêu cầu ghi rõ lý do close task.

Quy trình làm việc của Tester



b. Designer

- Thảo luận với ScrumMaster hoặc PO về yêu cầu thiết kế
- Cung cấp đủ và đúng hạn các thiết kế đồ họa cho nhóm phát triển
- Tuân thủ quy trình log task khi thiết kế cho dự án
 - Estimate time: ước lượng công việc đủ cho một task, không quá 3 giờ làm việc, nếu quá thì yêu cầu chia nhỏ công việc
 - Start date: Ngày bắt đầu công việc.
 - Due date: Ngày kết thúc công việc.
 - Status: Đổi trạng thái sang in-process.
 - Time logger: Bấm start time khi bắt đầu làm. Lưu ý: Có thể bấm start time khi công việc đó có trạng thái là NEW. Hệ thống sẽ tự chuyển sang trạng thái in-process khi bấm start time.

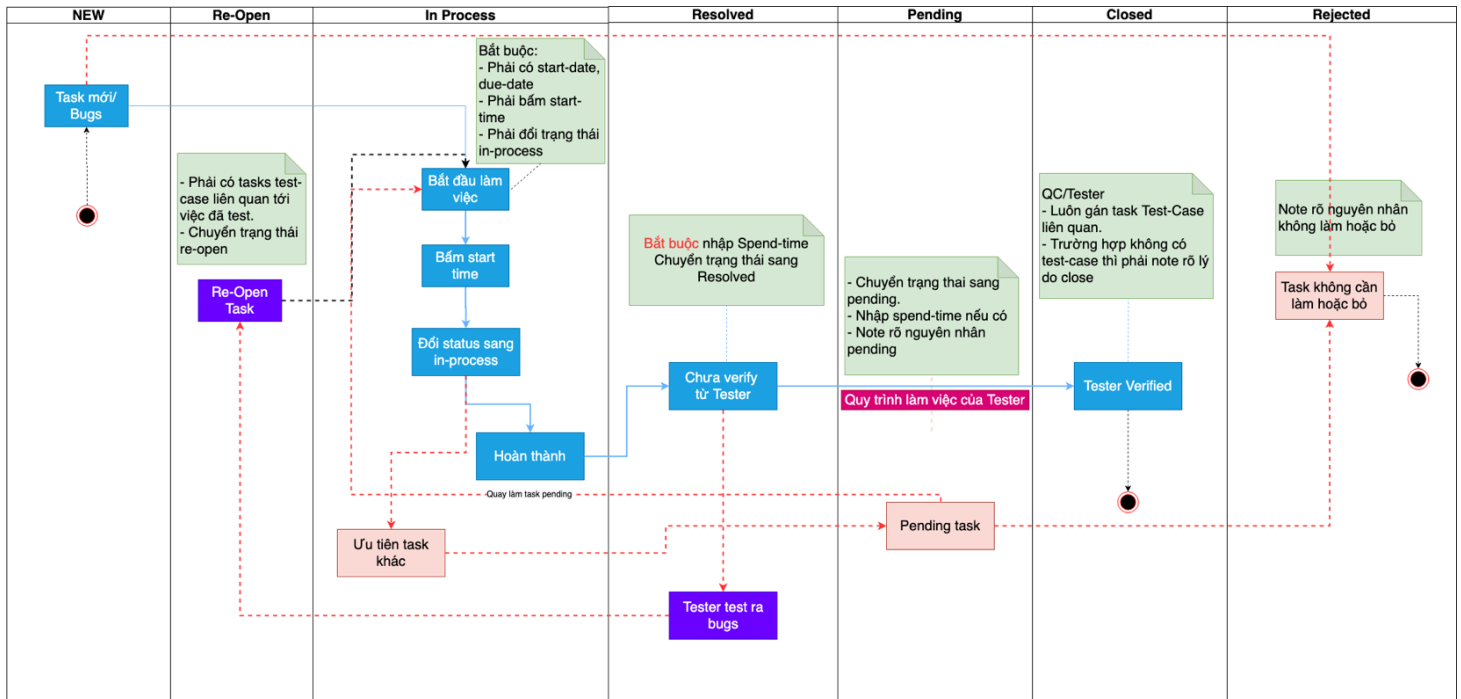
c. Business Analytics

- Phối hợp với phòng Kinh doanh và Khách hàng để làm rõ yêu cầu dự án trong giai đoạn pre-sales
- Hệ thống hoá yêu cầu dự án dưới dạng văn bản.

- <Lưu hành nội bộ> -

- Bàn giao yêu cầu dự án cho ScrumMaster trước khi dự án triển khai.
- Giải đáp thắc mắc cho ScrumMaster trong quá trình phát triển dự án.
- Hỗ trợ trao đổi thông tin giữa ScrumMaster và Khách hàng nếu cần.

II. CÁC BƯỚC PHÁT TRIỂN DỰ ÁN THEO SCRUM



1. Lập kế hoạch Sprint

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 có khả năng chuyển giao được.


Một dự án được chia thành nhiều Sprint bằng nhau. Sprint được đóng khung thời gian, thông thường có độ dài 2 tuần. Các Sprint diễn ra liên tiếp nhau mà không bị gián đoạn. Trong suốt Sprint:

- Không cho phép bất kỳ sự thay đổi nào ảnh hưởng đến Mục tiêu Sprint;
- Thành phần Nhóm Phát triển được giữ nguyên;
- Mục tiêu chất lượng không được cắt giảm; và,
- Phạm vi có thể được làm rõ và tái thương lượng giữa Product Owner và Nhóm Phát triển.

a. Lập kế hoạch

Là sự kiện diễn ra ở đầu mỗi Sprint để chuẩn bị cho toàn bộ Sprint. Buổi Lập kế hoạch Sprint được chia làm 2 phần riêng biệt với 2 mục đích khác nhau.

- Phần 1 nhằm trả lời câu hỏi: “Chúng ta sẽ làm gì?”
 - Product Owner trình bày cho Nhóm Phát triển hiểu rõ tất cả các hạng mục Product Backlog có khả năng được đưa vào sản xuất trong Sprint này. Thông thường, vì việc làm mịn các hạng mục Product Backlog đã được thực hiện liên tục trước đó cho nên lúc này Product Owner chỉ cần làm rõ một số hạng mục còn lại và trả lời các thắc mắc nếu có của Nhóm Phát triển.
 - Nhóm Phát triển lựa chọn các hạng mục Product Backlog để phát triển trong Sprint này. Số lượng hạng mục được lựa chọn hoàn toàn phụ thuộc vào Nhóm Phát triển, có thể dựa vào tốc độ sản xuất của nhóm trong quá khứ để ước lượng.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 12/16

- Kết thúc phần 1, Product Owner và Nhóm Phát triển đưa ra một Mục tiêu Sprint, Nhóm Phát triển cũng đã biết được tổng quan về công việc mà mình sẽ thực hiện trong Sprint này.
- Phần 2 nhằm trả lời câu hỏi: “Chúng ta sẽ làm như thế nào?”
 - Nhóm Phát triển phân tích và lên kế hoạch công việc cho Sprint.
 - Nhóm Phát triển bắt đầu bằng việc phân tách các hạng mục Product Backlog thành danh sách các công việc cụ thể.
 - Nhóm Phát triển cũng ước tính lượng nỗ lực cần bỏ ra để hoàn thành từng công việc.
 - Sau khi đã phân tích chi tiết, nếu Nhóm Phát triển thấy cần điều chỉnh (ví dụ, loại bớt một vài hạng mục Product Backlog đã chọn) thì có thể trao đổi với Product Owner để thực hiện thay đổi danh sách các hạng mục Product Backlog đã chọn.
 - Kết thúc phiên này, Nhóm Phát triển có được Sprint Backlog.

Kết thúc buổi Lập kế hoạch Sprint, Nhóm Phát triển đã biết Mục tiêu Sprint và kế hoạch công việc để đạt được mục tiêu đó. Các thành viên có thể bắt tay ngay vào công tác sản xuất.

b. Thành phần tham dự

Nhóm phát triển và ScrumMaster bắt buộc phải tham gia toàn bộ 2 phần của sự kiện này.

Product Owner bắt buộc phải tham gia Phần 1 và có thể vắng mặt ở Phần 2 nhưng phải đảm bảo sẵn sàng trả lời các thắc mắc của Nhóm Phát triển.

c. Thời gian

Đối với Sprint kéo dài 2 tuần thì sự kiện này được đóng khung trong 4 giờ làm việc. Sprint ngắn hơn thì thường sự kiện này sẽ ngắn hơn.


2. Scrum hàng ngày (daily meeting)

Scrum Hằng ngày là buổi trao đổi ngắn mà Nhóm Phát triển thực hiện đều đặn hằng ngày nhằm cập nhật và đồng bộ công việc giữa các thành viên.

- a. Thành phần tham dự: Tất cả các thành viên của Nhóm Phát triển bắt buộc phải tham gia sự kiện Scrum Hằng ngày. ScrumMaster không bắt buộc phải tham dự nhưng phải đảm bảo Nhóm Phát triển thực hiện tốt sự kiện này.
- b. Thời gian: tối đa hằng ngày là 15 phút, bất kể số lượng thành viên của Nhóm Phát triển là bao nhiêu.
- c. Mục đích của buổi Scrum Hằng ngày không phải là để giải quyết vấn đề mà là để thông báo và cập nhật tình hình. Do vậy, không cho phép bất cứ cuộc thảo luận chi tiết nào ở đây. Để đảm bảo tính ngắn gọn, lần lượt các thành viên của Nhóm Phát triển trình bày câu trả lời cho 3 câu hỏi:
 - Tôi đã làm gì từ buổi Scrum Hằng ngày trước cho đến bây giờ?
 - Tôi sẽ làm gì từ bây giờ cho tới buổi Scrum Hằng ngày hôm sau?
 - Tôi đang gặp phải những khó khăn gì?

Nếu có các vấn đề đòi hỏi phải thảo luận sâu giữa các thành viên thì Nhóm Phát triển sẽ tổ chức các buổi thảo luận ngay sau khi sự kiện Scrum Hằng ngày kết thúc.

Một thành viên của Nhóm Phát triển (thông thường là Tester) sẽ ghi lại các trở ngại để ScrumMaster hỗ trợ giải quyết sau khi buổi trao đổi kết thúc.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 13/16

3. Sơ kết Sprint

Sơ kết Sprint là sự kiện diễn ra ở cuối Sprint nhằm thanh tra và thích nghi sản phẩm đang được xây dựng. Sự kiện này bao 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.

- Thành phần tham dự: Nhóm Phát triển, ScrumMaster và Product Owner bắt buộc phải tham dự. Có thể mời những người khác như người dùng, khách hàng và các bên liên quan khác.
- Thời gian: tối đa trong 2 giờ đối với Sprint 2 tuần. Với các Sprint ngắn hơn thì cần ít thời gian hơn.
- Bắt đầu sự kiện, Product Owner sẽ trình bày về những hạng mục đã được lựa chọn cho Sprint, liệu chúng đã được hoàn thành hay chưa. Nhóm Phát triển có thể trình bày về những khó khăn mà mình gặp phải trong suốt Sprint và các giải pháp mà mình đã đưa ra.

Tiếp theo là buổi dùng thử sản phẩm: sắp xếp để những người tham gia có thể trực tiếp dùng thử sản phẩm, đặc biệt là những người dùng thực sự.

Sau đó, tất cả mọi người tham gia sẽ thảo luận và đóng góp ý kiến cho sản phẩm. Product Owner và Nhóm Phát triển ghi nhận những ý kiến này. Product Backlog và Kế hoạch Phát hành có thể được thay đổi nếu cần thiết để phù hợp hơn với tình hình mới.

4. Cải tiến Sprint

Cải tiến Sprint diễn ra ngay sau buổi Sơ kết Sprint nhằm mục đích thanh tra và cải tiến quy trình làm việc.

- Thành phần tham dự: Nhóm Phát triển và ScrumMaster bắt buộc phải tham dự. Product Owner có thể tham dự hoặc không. Có thể mời thêm những người khác cùng tham dự nếu cần thiết.
- Thời gian: Sự kiện này được đóng khung trong 1,5 giờ đối với Sprint 2 tuần. Với các Sprint ngắn hơn thì thời gian có thể ngắn hơn, vào khoảng 45 phút tương ứng với 1 tuần làm việc của Sprint.
- Mục đích của buổi Cải tiến Sprint bao gồm:
 - Thanh tra lại Sprint trước, về các yếu tố liên quan đến con người, giao tiếp, quy trình và công cụ.
 - Liệt kê những hạng mục đã làm tốt và những hạng mục có thể cải tiến được.
 - Lên kế hoạch triển khai các cải tiến về cách làm việc của Nhóm Scrum..


III. LƯU Ý KHÁC

1. Version

ScrumMaster và QC chịu trách nhiệm quản lý Sprint và Version.

Phiên bản release version là tổng hợp các chức năng gói gọn gửi tới khách hàng. Version có thể kéo dài tới 1 tháng, 1 tuần, trong version có thể chia làm nhiều sprint, Đôi khi thời gian kết thúc và bắt đầu bằng 1 sprint. EX: sprint 1 là phát triển cho version 1.0.1 trong thời gian 1 tuần. version 1.0.11 là version dọn sprint cho sprint 1. Sprint 2 phát triển cho version 1.0.2

- Quy ước đặt tên phiên bản [tên gọi nhớ] major.minor[.build[.revision]] trong đó:
 - [tên gọi nhớ] dùng để nhóm các loại công việc được thực hiện trong 1 giai đoạn hoặc nhóm các chức năng được phát triển trên 1 nền tảng công nghệ, hoặc tên của một module/component được phát triển.
 - major.minor[.build[.revision]] tham khảo quy ước định dạng phiên bản.
- Quy ước định dạng phiên bản major.minor[.build[.revision]]. Trong đó:

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 14/16

- Major: là số để đánh dấu có sự thay đổi lớn về chức năng, như thay đổi lõi framework, thay đổi có ảnh hưởng đến khả năng tương thích các chức năng đã có của hệ thống hay thêm mới một loạt tính năng mới.
- Minor: là số để chỉ những thay đổi nhỏ về tính năng hoặc những cập nhật để vá các lỗi lớn.
- Build: là số không bắt buộc, tuy nhiên số này có thể sử dụng để chỉ ra một số trạng thái của phiên bản phần mềm được release. Các trạng thái được quy định với các con số như:
 - chỉ phiên bản alpha. Ví dụ: 1.2.0
 - 1 chỉ phiên bản beta. Ví dụ: 1.2.1
 - 2 chỉ phiên bản release candidate. Ví dụ: 1.2.2
 - 3 chỉ phiên bản final. Ví dụ: 1.2.3
- Revision: là số không bắt buộc, mục đích chủ yếu để chỉ những số lần commit lên hệ thống quản lý mã nguồn như SVN, Git. Số này thường tăng lên khi fix các bug nhỏ. Thường thì số revision đi chung với số build để chỉ rõ những thay đổi nhỏ trong từng bản build. Khi release 1 component hoặc 1 module để gọi nhờ các thay đổi sau cùng, revision được sử dụng để ghi lại cùng danh sách các thay đổi ứng với nội dung trong file change log hoặc release note: EX:
 - 1.2.0.112 chỉ phiên bản alpha đã cập nhật và lỗi tại revision 112.
 - 1.2.1.244 chỉ phiên bản beta đã cập nhật và lỗi tại revision 244

2. Product Backlog

Product Backlog là nơi lưu trữ danh sách các tính năng mong muốn của sản phẩm. Danh sách này được sắp xếp dựa trên độ ưu tiên của từng hạng mục. Các hạng mục có độ ưu tiên cao hơn nằm ở phía trên của danh sách và sẽ được Nhóm Phát triển lựa chọn để đưa vào sản xuất sớm, các hạng mục có độ ưu tiên thấp hơn sẽ nằm ở phía cuối của danh sách và được phát triển muộn hơn.

Product Backlog có thể chứa các hạng mục thuộc các loại như:

- Tính năng sản phẩm
- Lỗi
- Công việc liên quan đến kỹ thuật
- Công việc nghiên cứu

Một Hạng mục Product Backlog có thể được mô tả dưới bất cứ hình thức nào, chẳng hạn như: User Story, User Case, User Scenario,...

Một Product Backlog tốt cần thỏa mãn tiêu chí DEEP, bao gồm:

- Detailed Appropriately (Đủ chi tiết Hợp lý): Có nghĩa là các hạng mục cần có đủ chi tiết để hiểu đúng, nhưng không có nghĩa là tất cả các hạng mục đều có một mức độ chi tiết như nhau mà chúng được giữ ở mức chi tiết phù hợp. Có nghĩa là, những hạng mục ở trên cùng (sẽ được đưa vào sản xuất sớm) cần có đầy đủ chi tiết ở mức cao nhất, những hạng mục ở phía dưới (sẽ được đưa vào sản xuất muộn hơn) thì chỉ cần giữ ở mức chi tiết cơ bản.
- Estimated (Được ước tính): Tất cả các hạng mục đều phải được ước tính, bao gồm cả giá trị thương mại và kích thước (lượng nỗ lực cần thiết để xây dựng).
- Emergent (Tiến hóa): Product Backlog không phải là một danh sách cố định. Nó liên tục được cập nhật và duy trì trong suốt quá trình phát triển dựa theo những hiểu biết học được.

 Adamo Digital	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 15/16

- **Prioritized (Sắp xếp theo độ ưu tiên):** Các hạng mục trong Product Backlog cần được sắp xếp theo độ ưu tiên để tối ưu hóa giá trị của công việc phát triển.

Product Backlog là duy nhất, kể cả trong trường hợp có một hay nhiều Nhóm Scrum cùng làm tham gia phát triển sản phẩm.

3. Làm mịn Product Backlog

Việc làm mịn Product Backlog là hoạt động thêm vào các chi tiết, ước lượng, và trình tự của các hạng mục trong Product Backlog. Đây là quá trình liên tục, theo đó Product Owner và Nhóm Phát triển thảo luận về các chi tiết của từng hạng mục. Trong suốt quá trình làm mịn này, các hạng mục liên tục được xem xét và rà soát cẩn thận.

Nhóm Scrum quyết định cách thức và thời điểm để làm mịn Product Backlog. Có những nhóm có hoạt động làm mịn Product Backlog là một sự kiện diễn ra vào giữa Sprint để chuẩn bị yêu cầu cho sprint tiếp theo. Hoạt động làm mịn product backlog không chiếm nhiều hơn 10% thời gian của Nhóm Phát triển. Tuy thế, các hạng mục Product Backlog có thể được cập nhật tại bất kì thời điểm nào theo chủ quan của Product Owner.

4. User Stories

User Story là một tài liệu sơ giản về yêu cầu sản phẩm với góc nhìn người dùng. Thông thường, User Story do khách hàng, hoặc đại diện của khách hàng viết, tuy nhiên nếu có sự cộng tác của Nhóm Phát triển thì nhóm và khách hàng sẽ có sự chia sẻ hiểu biết về sản phẩm tốt hơn.

User story có định dạng:


Là <người dùng cụ thể/vai trò> ,
tôi muốn <làm gì đó>
để <phục vụ mục đích nào đó>

User story nên theo mô hình 3C:

- **Card (Thẻ):** Thông thường, User Story được viết trên một thẻ nhỏ. Điều đó có nghĩa là nó thường ngắn để có thể viết trên một thẻ. Nếu bạn có viết trên một hệ thống khác như Trello, Jira, Assembla hoặc Redmine cũng nên giữ nó ngắn.
- **Conversation (Trao đổi):** Story là những câu truyện giữa khách hàng và Nhóm Phát triển. Do đó chi tiết của User Story được làm rõ thông qua các cuộc trao đổi (nên là trực tiếp) với khách hàng. Nội dung của User Story sẽ ngày càng cụ thể tùy thuộc vào độ ưu tiên của nó (nếu ưu tiên cao, cần làm sớm thì sẽ có nội dung chi tiết, nếu ưu tiên thấp thì chỉ chứa nội dung chung).
- **Confirmation (Xác nhận):** User Story có tiêu chí chấp nhận (Acceptance Criteria) để khách hàng suy nghĩ cụ thể về yêu cầu và Nhóm Phát triển có thể hiểu yêu cầu rõ hơn và xác nhận được khi nào sản phẩm hoàn thành.

User story cũng nên tuân theo các tiêu INVEST:

- **Independent (Độc lập):** Độc lập với các User Story khác. Điều này giúp Product Owner tự do thay đổi thứ tự của nó trong Product Backlog và Nhóm Phát triển dễ dàng phát triển.
- **Negotiable (Thương lượng được):** Tính đàm phán được giúp cho Nhóm Phát triển và Product Owner cùng nhau xây dựng nội dung chi tiết và phụ hợp hơn cho những thay đổi trong tương lai. Nếu không có tính năng này thì việc thích nghi với sự thay đổi gặp khó khăn.

	QUY TRÌNH PHÁT TRIỂN PHẦN MỀM		
Mã: ADM-DEV-001	Phiên bản: 1.0	Hiệu lực: 01/06/2020	Trang: 16/16

- **Valuable (Có giá trị):** User Story phải có giá trị với khách hàng. Những người làm kỹ thuật có thể thấy việc làm khung làm việc, cơ sở dữ liệu hoặc thiết kế là quan trọng. Tuy nhiên với khách hàng thì không. Điều này rất lưu ý với những Product Owner có nền tảng kỹ thuật, có thể họ sẽ biết Agile thành một mô hình phát triển Waterfall trá hình!
- **Estimable (Ước lượng được):** Một User Story tốt có thể ước lượng được mặc dù không cần chính xác. Những User Story lớn hoặc không rõ ràng thường khó để ước lượng. Khả năng ước tính được giúp nhóm ước lượng tốt hơn công việc sẽ làm và cả kế hoạch phát hành. Rõ ràng điều này phụ thuộc vào khả năng của nhóm.
- **Sized appropriately (Kích thước phù hợp):** Những User Story sắp được đưa vào sản xuất cần có kích thước nhỏ (đồng nghĩa với việc được mô tả rõ ràng hơn), những User Story chưa được đưa vào sản xuất trước mắt có thể có kích thước lớn hơn.
- **Testable (Kiểm thử được):** Nếu nhóm phát triển biết như thế nào là User Story đó hoàn thành – có thể kiểm thử được rõ ràng thì họ có thể hiểu rõ hơn công việc của mình, ít gây hiểu nhầm. Các mô hình phát triển BDD hoặc ATDD có giá trị vì yêu cầu có thể kiểm thử được.

5. Định nghĩa hoàn thành

Thành quả của Nhóm Phát triển sau mỗi Sprint là Phần tăng trưởng Sản phẩm, Có khả năng Chuyển giao được. Tức là những hạng mục trong Product Backlog đã “Hoàn thành”. Như vậy khái niệm “Hoàn thành” được hiểu như sau:

- Tính năng chạy được
- Mã nguồn được kiểm tra chéo theo tiêu chuẩn mã nguồn
- Vượt qua kiểm thử chấp nhận người dùng
- Vượt qua tất cả kiểm thử hồi quy

6. Meeting minutes

Trong mỗi cuộc họp: (Sprint, Daily meeting): một thành viên Nhóm phát triển (thường là Tester) cần:

- luôn tạo task cuộc họp trong dự án trước khi bắt đầu
- ghi lại nội dung chính của cuộc họp
- assign chính xác cho người tổ chức cuộc họp, tag những người liên quan vào cuộc họp.

IV. HỒ SƠ

TT	Tên hồ sơ	Mã hiệu	Phương thức lưu		Thời gian lưu (năm)	Bộ phận lưu
			Hard	Soft		
1	Product backlog			✓	5	DEV
2	Meeting minutes			✓	5	DEV
3	Test cases			✓	5	DEV
4	Source-code			✓	5	DEV