

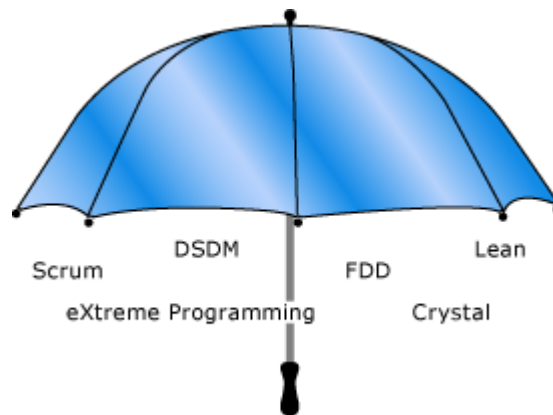
Buổi 1:

Quy trình của dự án là gì?

Nhu cầu thực sự của người dùng khi yêu cầu chức năng thường là rất lớn và sẽ có những chức năng không dùng đến.

Agile software development hay Agile programming Phát triển phần mềm linh hoạt hoặc Lập trình linh hoạt – là tập hợp các nguyên lý của các mô hình phát triển phần mềm linh hoạt và nhanh chóng. Trên thế giới chúng ta thấy có nhiều xuất hiện nhiều phương pháp phát triển phần mềm mà **có nguyên tắc** của Agile đó là mô hình **Extreme programming (XP) is a software development methodology (được ứng dụng ở Microsoft)**, **Lean software development (Japan)**, **Scrum** – được nghe đến và sử dụng khác phổ biến hiện nay....

Agile là chiếc ô – Các phương pháp được triển khai dưới chiếc ô này



Tuyên ngôn của **Agile**

Tuyên ngôn Agile

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

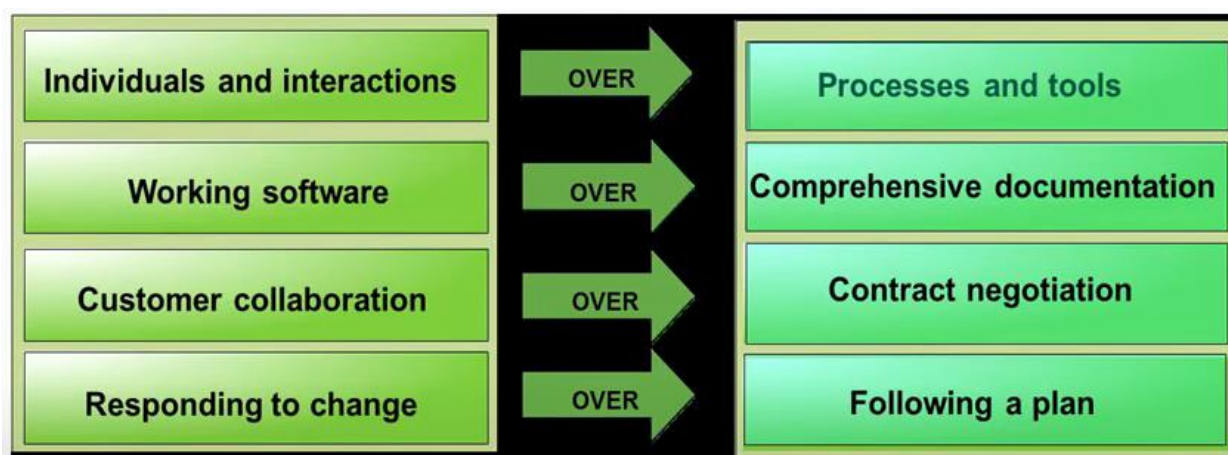
Cá nhân và sự tương tác hơn là quy trình và công cụ

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.



Khi một mô hình phần mềm nào đó **thỏa mãn 4 nguyên tắc** này thì chúng ta gọi đó là phương pháp phát triển phần mềm **Agile**. Trong Agile thì sẽ có nhiều hình thức triển khai dự án và trong đó chúng ta thấy Scrum là một trong những mô hình phù hợp để ràng áp dụng trong tất cả các dự án khác. Lean có những đặc thù của người nhật và XP có đặc thù riêng của Microsoft và đôi khi khó áp dụng ra bên ngoài nhưng riêng với Scrum có thể nói là với tất cả dự án đều phù hợp.

Loại dự án phù hợp với mô hình Agile - Scrum

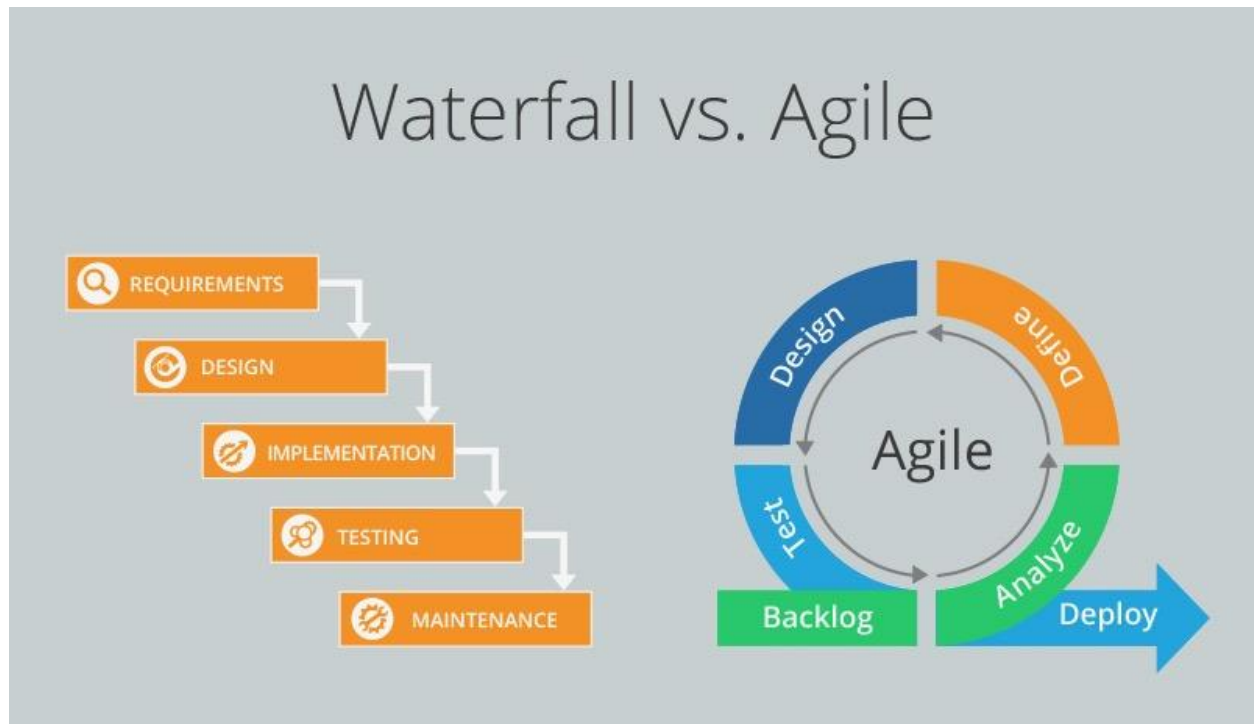
Khách hàng phải cam kết tham gia vào quá trình của dự án về mặt thời gian và phê duyệt yêu cầu thường xuyên và liên tục. Yêu cầu của dự án chưa có những chuẩn và những quy định rõ ràng cần có sự nghiên cứu – sáng tạo – thử nghiệm.

Loại dự án không phù hợp với mô hình Agile - Scrum

Với dự án đã có những nguyên tắc cố định ví dụ như phần mềm kế toán – ngân hàng và có những luật định rõ ràng thì ko nên sử dụng mô hình Agile Scrum. Ví dụ các phần mềm về kế toán và phần mềm về quy trình nghiệp vụ của ngân hàng, đây là những phần mềm rất nguyên tắc và có những nghiệp vụ chắc

chắc không thay đổi thì bạn không cần sáng tạo mà chỉ cần làm theo đúng nguyên tắc đã đề ra là được chỉ cần dùng mô hình cũ như waterfall là đủ.

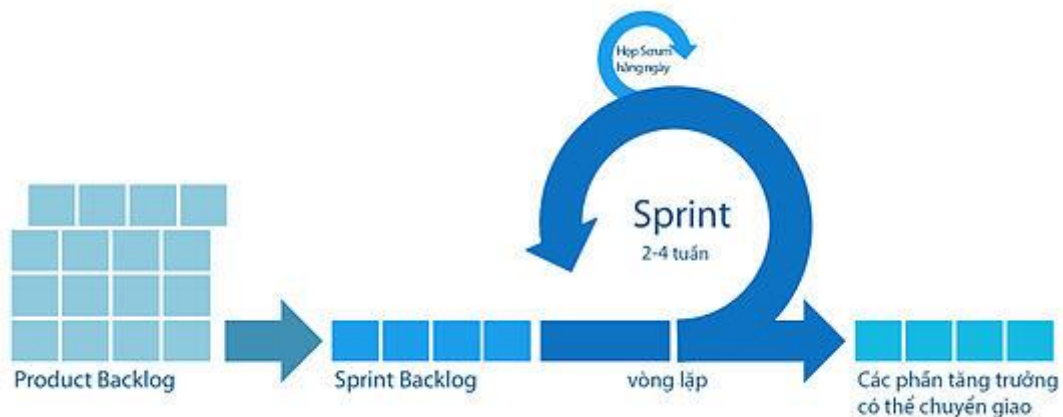
Mô hình Agile



Thế nào thì là Scrum?

Ra đời năm 1990 bởi Ken Schwaber và Jeff Sutherland với mục tiêu cung cấp một sản phẩm có giá trị ngày càng gia tăng trong những khoảng thời gian cố định được (Sprint).

1. **Scrum** hoạt động dựa trên những giai đoạn (Sprint), thường kéo dài từ **2 đến 4 tuần**.
2. **Scrum** là mô hình thúc đẩy sự tương tác và gia tăng hoạt động tự quản lý dự án dựa trên việc tổ chức một nhóm có khả năng chuyên sâu làm việc phụ thuộc lẫn nhau trong một khoảng thời gian cố định.



Khung làm việc của Scrum

1. **Product backlog:** Tập hợp tất cả các yêu cầu của một sản phẩm và các yêu cầu được viết dưới dạng **User Story** tức là với ngôn ngữ của người sử dụng hay còn gọi là câu chuyện của người sử dụng – người sử dụng mong muốn gì. **Product backlog** là tập hợp toàn bộ các yêu cầu của sản phẩm tuy nhiên nó luôn được sắp xếp theo thứ tự ưu tiên theo những **User Story** nào có thứ tự ưu tiên cao nhất chúng ta sẽ triển khai trước. Những **User Story** có thứ tự ưu tiên thấp có thể để lại sau và có khi chúng ta sẽ bỏ qua nó không cần triển khai nữa sau khi người sử dụng đã thỏa mãn với những tính năng đã có của phần mềm vì thế **Product backlog** luôn được sắp xếp và được phân thứ tự ưu tiên một cách thường xuyên để chúng ta loại đi những **User Story** không cần thiết và đưa ra những **User Story** cần thiết nhất lên trên đầu.
2. **Sprint backlog:** Tập hợp các **User Story** sẽ làm trong 1 **Sprint**. 1 Sprint chỉ tiến hành từ 2 đến 4 tuần vì thế chúng ta sẽ lựa chọn các User Story chọn ra và phải cam kết hoàn thành được các User Story đã lựa chọn. Như vậy những User Story lựa chọn chúng ta đưa vào Sprint chúng ta gọi đó là các Sprint Backlog.
3. Trong quá trình thực hiện **Sprint** chúng ta sẽ tổ chức một buổi họp **Sprint Planning** buổi họp sẽ có tất cả các thành viên của đội dự án và có cả đại diện của cả khách hàng để trả lời các **Q&A** của đội dự án. Đội dự án sẽ thảo luận và đưa ra các **Task** công việc cần làm trong một **Sprint**.
4. **Daily meetings:** Hằng ngày đội dự án sẽ có cuộc họp gọi là **Daily meeting** và quan sát **Burn Down Chart** để biết được tiến độ của dự án.
5. Khi kết thúc 1 **Sprint** chúng ta sẽ có một buổi họp **demo** cho khách hàng và rút kinh nghiệm và tiếp tục lại tổ chức một **Sprint** tiếp theo. Cứ như vậy các **Sprint** tổ chức liên tục cho đến khi khách hàng cảm thấy thỏa mãn với tất cả các tính năng của phần mềm và họ không muốn xây dựng bất cứ tính năng nào nữa thì chúng ta sẽ kết thúc dự án và bàn giao toàn bộ sản phẩm cho khách hàng.

Các vai trò trong Scrum (Role)

1. **Product Owner:** là người đại diện của khách hàng có vai trò và tiếng nói quyết định về các tính năng cũng như các yêu cầu thực hiện trong dự án.
2. **Scrum Master:** là người thực hiện các quy trình của Scrum trong dự án.
3. **Team:** Là các thành viên của dự án bao gồm **BA – DEV – TESTER - QA**

Artifact

1. **Definition of done:** Định nghĩa tiêu chí về thế nào là độ hoàn thành. Chúng ta phải thống nhất với nhau về các tiêu chí hoàn thành cho một dự án cho một Sprint, cho một sản phẩm, 1 task.
2. **Product backlog:** là một tập hợp quản lý các yêu cầu của dự án xuyên suốt từ đầu cho đến cuối dự án **Product backlog** sẽ do Product Owner là người tạo ra và quản lý sắp xếp thứ tự ưu tiên. **Product backlog** sẽ được cập nhật liên tục hằng ngày và hàng Sprint.
3. **Release Plan:** cũng do **Product Owner** lập ra và đây là kế hoạch bàn giao sản phẩm. Chúng ta sẽ có những chặng bàn giao để kịp tiến độ với khách hàng.
4. **Sprint backlog:** là các yêu cầu để làm việc trong từng sprint.
5. **Kanban Board:** đây là một bảng theo dõi và quản lý **Task** công việc của một dự án.
6. **Velocity:** Chỉ số năng suất trên một dự án chỉ số này được đo bằng **User Story** point trên một Sprint.

Sự kiện: Event

1. **Sprint Planning:** Là buổi lập kế hoạch trong một Sprint
2. **Daily Meeting:** là buổi họp hằng ngày cho các thành viên trong đội dự án để theo dõi và báo cáo kết quả của toàn đội dự án.
3. **Sprint Review:** hay là sprint meeting là buổi chúng ta demo giới thiệu kết quả của 1 sprint và lấy yêu cầu và mong muốn thay đổi của khách hàng trong buổi Sprint Review.
4. **Sprint Retrospective:** là buổi họp rút kinh nghiệm của toàn đội dự án cho các Sprint tiếp theo và đẩy các sprint tiếp theo lên nhanh hơn.

Tôn Trọng các Giá trị của Scrum khi tham gia dự án

1. **Commitment:** Tính cam kết cao trong công việc
2. **Focus:** Tập trung vào đúng mục tiêu, chiến lược trong một thời điểm nhất định.
3. **Openness:** Mọi thông tin đều minh bạch đối với cả trong và ngoài dự án.
4. **Respect:** Giá trị gắn kết giữa các thành viên dự án với nhau và với khách hàng cũng như công ty.
5. **Courage:** Giá trị của sự can đảm, dám nhận và cam kết với công việc đề ra cho dù là các task khó hơn, dũng cảm công khai minh bạch mọi thông tin.

Trọng tâm của Scrum

1. **Timeboxing:** Các quy tắc về thời gian luôn luôn là cố định. Khi chúng ta đưa ra 1 Sprint 2 tuần thì đúng 2 tuần chúng ta phải kết thúc 1 Sprint không bao giờ kết thúc sớm hoặc muộn. Bằng mọi giá để có thể kết thúc 1 Sprint. Có thể bỏ bớt Task đi hoặc nếu xong sớm chúng ta có thể bổ sung Task công việc vào Sprint cho đến khi Sprint kết thúc.
 - **Sprint Planning** trong vòng 4 tiếng đồng hồ chúng ta phải lập kế hoạch thật tập trung để làm sao lập kế hoạch trong đúng 4 tiếng.
 - **Daily Meeting** chỉ được diễn ra trong vòng 15 phút.
2. **Learning from mistakes and self managing:** Học hỏi và rút kinh nghiệm từ những sai lầm và sự tự chủ động quản lý.
3. **Shippable Code:** sản phẩm phải chạy hay hoạt động được là trọng tâm của dự án. Các yêu cầu phải được làm và hoàn thiện gọn gàng để có thể thực hiện được hoàn chỉnh và sản phẩm được đóng gói thành nhiều lần theo từng giai đoạn để sử dụng thử và cải tiến.
4. **Enough to Start:** Vừa đủ để thực hiện, các hoạt động của Scrum được diễn ra khi thông tin vừa đủ để bắt đầu, không chờ đợi đến khi mọi thứ được rõ ràng và đầy đủ mới bắt đầu thực hiện. Các yêu cầu chỉ nên mô tả ở dạng ngắn gọn và khi bạn cần đến bạn phải tìm hiểu và thảo luận nó và trao đổi với **Product Owner** để biết rõ về Task công việc của mình như là sẽ có thể các buổi họp với khách hàng. Không bao giờ chúng ta thấy được 1 tài liệu rõ ràng đầy đủ vì vậy Scrum luôn đòi hỏi yêu cầu sự tương tác giữa các nhân và khách hàng cao hơn tất cả về quy trình và công cụ.

So sánh Scrum và Water Fall



Vai trò và trách nhiệm của Product Owner

1. Xác định lộ trình phát triển của sản phẩm kế hoạch bàn giao của sản phẩm
2. Xác định thứ tự ưu tiên cho các tính năng của sản phẩm.
3. Thiết lập các quy tắc trong việc phát triển sản phẩm.
4. Mô tả và duy trì mức độ chi tiết cho các tính năng của sản phẩm.
5. Sẵn sàng chuẩn bị cho những đàm phán có thể xảy ra.
6. Định hướng nghiệp vụ trong quá trình phát triển sản phẩm
7. Chấp nhận và xác định mức độ có thể chấp nhận sản phẩm (Xác định thời điểm kết thúc việc phát triển sản phẩm)
8. Vị trí này có thể là: Trưởng phòng sản xuất, quản trị dự án.... Hoặc khách hàng.

Vai trò và trách nhiệm của Scrum Master(PM)

1. Bảo vệ nhóm và giữ cho họ tập trung tới mục tiêu đã xác định.

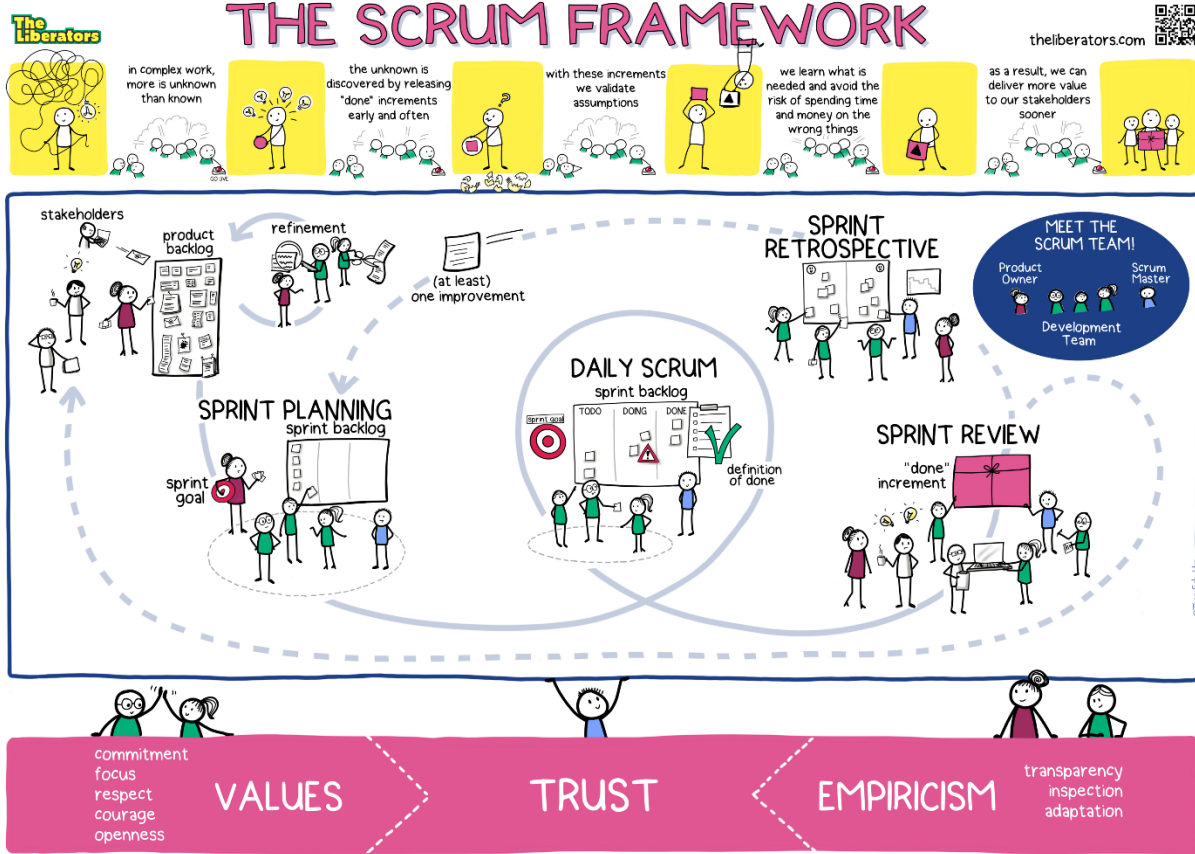
2. Loại bỏ các trở ngại đối với nhóm để hướng tới mục tiêu của Sprint
3. Chấp hành, thi hành các quy tắc Scrum đối với nhóm.
4. Là thành viên điều phối của nhóm.
5. Đại diện cho nhóm trước các bên liên quan.
6. SM phải đặt việc phục vụ nhóm lên hàng đầu.
7. Lắng nghe đồng đội cũng như các bên liên quan.
8. Phương pháp làm việc hướng tới việc tự tổ chức nhóm.
9. Luôn tạo điều kiện và loại bỏ các trở ngại.

Vai trò và trách nhiệm Team

1. **Làm việc đa năng trong dự án** (Có thể làm nhiều công việc khác nhau không phụ thuộc vào chuyên môn).
2. **Không phân cấp thứ bậc trong dự án** (Không quản lý ai)
3. **Tất cả cùng chịu trách nhiệm với sản phẩm bàn giao cho khách hàng.**
4. **Báo cáo tình hình công việc mình làm cho Scrum Master và cả Teams.**

Vai trò của Stakeholder (Customers, vendors)

1. Đây là những người cho phép dự án được triển khai và những người mà dự án phụ thuộc vào và soạn ra những thỏa thuận về lợi ích trong việc phát triển phần mềm trong dự án.
2. Những người này chỉ tham gia trực tiếp trong những buổi demo hoặc bàn giao sản phẩm để kiểm chứng khả năng thực tế của hệ thống phần mềm.
3. Managers: cũng tương tự là những người thiết lập/ xây dựng môi trường làm việc cho sự phát triển của sản phẩm và không tham gia trực tiếp vào dự án.



Buổi 2:

KANBAN BOARD (JAPAN) được sử dụng nguyên tắc (**Pull**) kéo đẩy là khi thực hiện xong 1 công việc chúng ta sẽ kéo sang làm một công việc khác. Không cho phép công việc được dư thừa tồn đọng và lúc nào cũng phải linh hoạt tiết kiệm. Hoàn toàn có thể bổ sung các cột cho một quy trình tương tự và mở rộng hơn tùy thuộc theo dự án. (Dùng giấy Note)

KANBAN BOARD				
US	TODO	IN PROGRESS	PENDING	DONE
User Story	Những Task công việc chưa làm đến	Những Task công việc đang được thực hiện	Những Task đang gặp phải (issue) vấn đề rắc rối không xử lý được.	Dành cho các Task đã hoàn thành. Đã xong đã kiểm tra xong.
Viết các US đưa vào 1 Sprint lên đây và giữ nguyên vị trí				
1 User Story	3 Task trên cùng 1 dòng			

KANBAN BOARD sẽ được gắn kèm với Agile Scrum khi bắt đầu 1 Sprint. Chúng ta sẽ đưa các US nào thực hiện lên một Sprint thì sẽ đưa lên Kanban Board.

Trên các giấy Note chúng ta sẽ viết rõ tên các User Story để viết rõ tên công việc cần phải làm là gì – trong buổi planning ngoài việc lựa chọn các US chúng ta sẽ tiến hành Break Task tức là chúng ta sẽ chia các công việc để làm và hoàn thiện 1 cái US vì thế chúng ta có thể có nhiều Task cho 1 cái User Story để làm ví dụ như GUI – Coding – Testing.

Hàng ngày các thành viên dự án sẽ chọn task công việc từ cột todo sang cột in progress thì mới bắt đầu viết tên chúng ta vào để biết ai là người đang thực hiện công việc đấy.

Lưu ý MỖI MỘT NGƯỜI CHỈ ĐƯỢC THỰC HIỆN MỘT TASK và chỉ được chuyển một TASK trong một thời điểm. KHI HOÀN THÀNH THÌ MỚI ĐƯỢC LẤY TASK TỪ CỘT TODO để làm tiếp. TRONG MỘT INPROGRESS KHÔNG THỂ CÓ 2 TASK DO MỘT NGƯỜI LÀM. NGUYÊN TẮC KHÔNG LÀM 2 ĐẾN 3 CÔNG VIỆC.

NẾU BẠN LÀM TASK ĐÓ MÀ CẢM THẦY KHÔNG PHÙ HỢP HÃY KÉO NÓ VỀ CỘT TODO để cho người khác lựa chọn và bạn chọn Task khác kéo về cột INP PROGRESS.

DỰA VÀO BẢNG KANBAN KHÔNG CẦN PHẢI HỎI CÁC THÀNH VIÊN TRONG DỰ ÁN MÀ CÓ THỂ ĐÁNH GIÁ TIẾN ĐỘ DỰ ÁN MỘT CÁCH TRỰC QUAN VÀ LINH HOẠT.

BURNDOWN CHART

Là một biết đồ quản lý tiến độ công việc của dự án Agile Scrum và quản lý theo tiến độ giảm dần.

Đầu tiên chúng ta sẽ đếm số lượng Task vào ngày đầu tiên khi làm plan.

Thứ hai là ngày cuối cùng của Sprint.

Hàng ngày khi Daily Meeting thì chúng ta sẽ vẽ lại đường burn down chart theo điểm thực tế và khi họp sẽ đếm số lượng task còn lại trên cột todo.

File này đã có trong tài liệu tham khảo.

Nguyên tắc SMART trong xác định mục tiêu

Nguyên tắc SMART là gì?

Nguyên tắc **SMART** còn gọi là nguyên tắc **THÔNG MINH** giúp định hình và nắm giữ mục tiêu của bạn trong tương lai; qua đó, bạn sẽ xác định những khả năng mà mình có thể làm được và xây dựng kế hoạch cụ thể để hiện thực hóa những mục tiêu đó.



Cách thiết lập mục tiêu Thông minh – Mô hình SMART

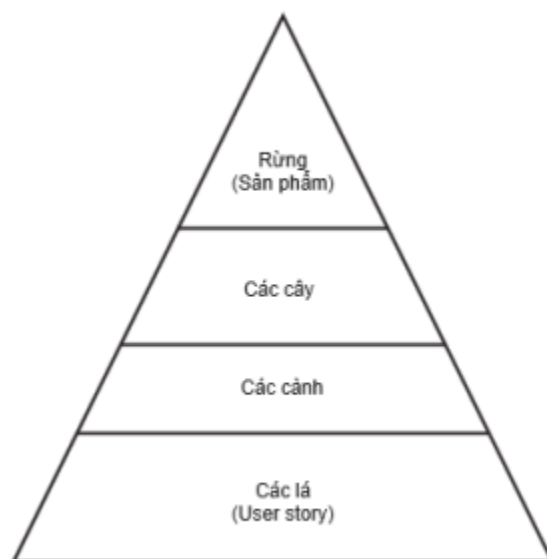


S pecific	<ul style="list-style-type: none"> • Cụ thể, rõ ràng và dễ hiểu • (Thức dậy vào lúc 5:00 mỗi buổi sáng)
M easurable	<ul style="list-style-type: none"> • Đo đếm được • (Uống 2 lít nước mỗi ngày)
A ttainable	<ul style="list-style-type: none"> • Có thể đạt được bằng sức của mình • (Chạy bộ 30km/h => không tưởng)
R elevant	<ul style="list-style-type: none"> • Liên quan đến tầm nhìn chung • (Học tiếng Pháp 2h/ngày => không phục vụ mục tiêu du học Anh)
T ime-bound	<ul style="list-style-type: none"> • Có thời hạn • (Biết chơi guitar trong vòng 1 tháng tới)

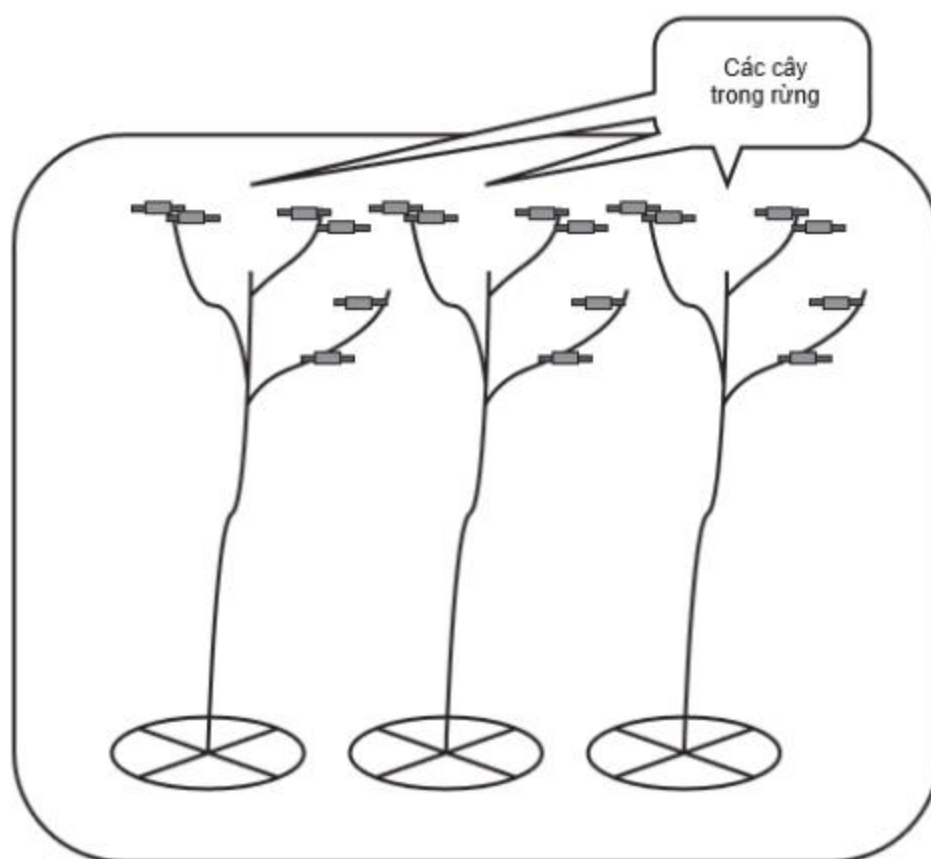
Thu thập yêu cầu cho Product Backlog

Trong bước này, bạn sẽ gặp những người đại diện cho các bên liên quan, mỗi người có vai trò riêng để hiểu những gì mà người dùng cần và chuyển thành các yêu cầu của sản phẩm mới

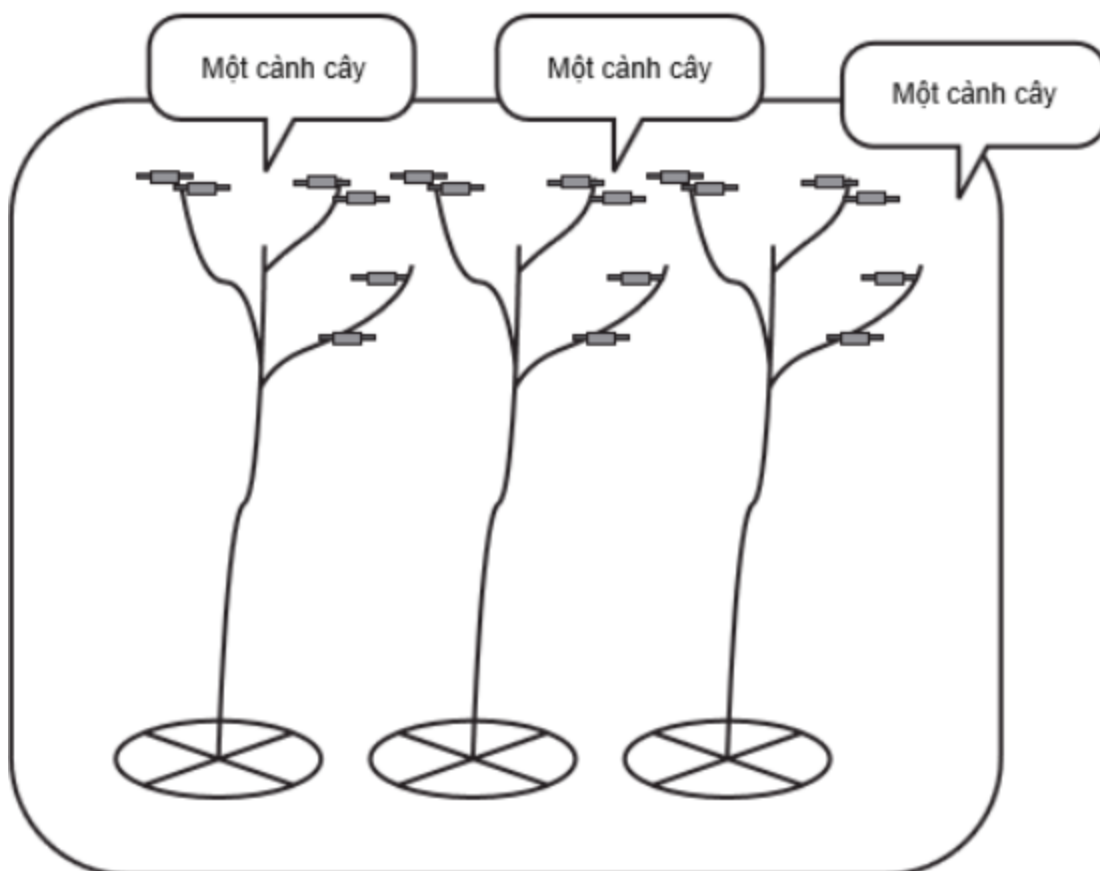
Sử dụng kỹ thuật “cây và rừng”



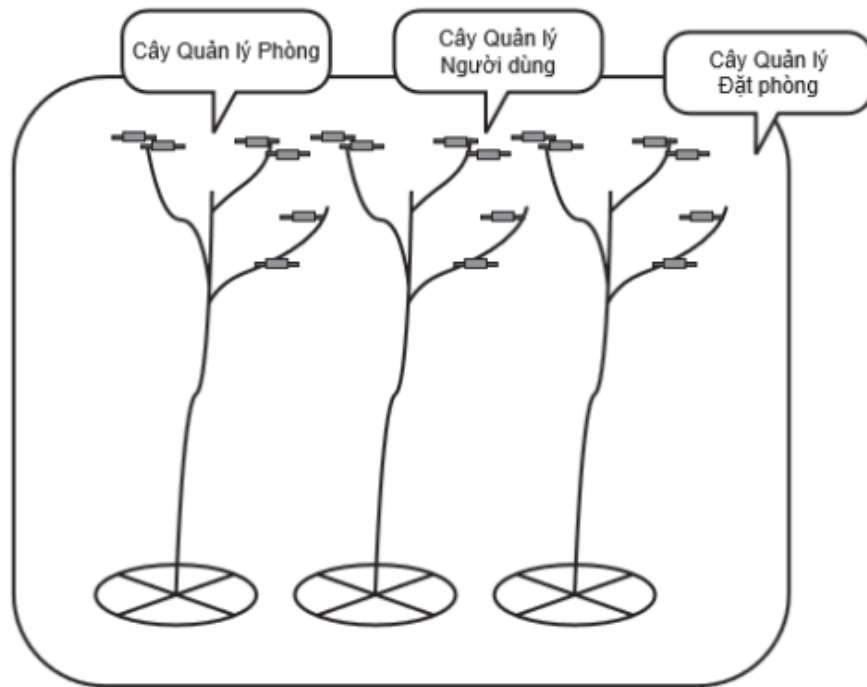
Kỹ thuật cây và rừng



Kỹ thuật cây và rừng

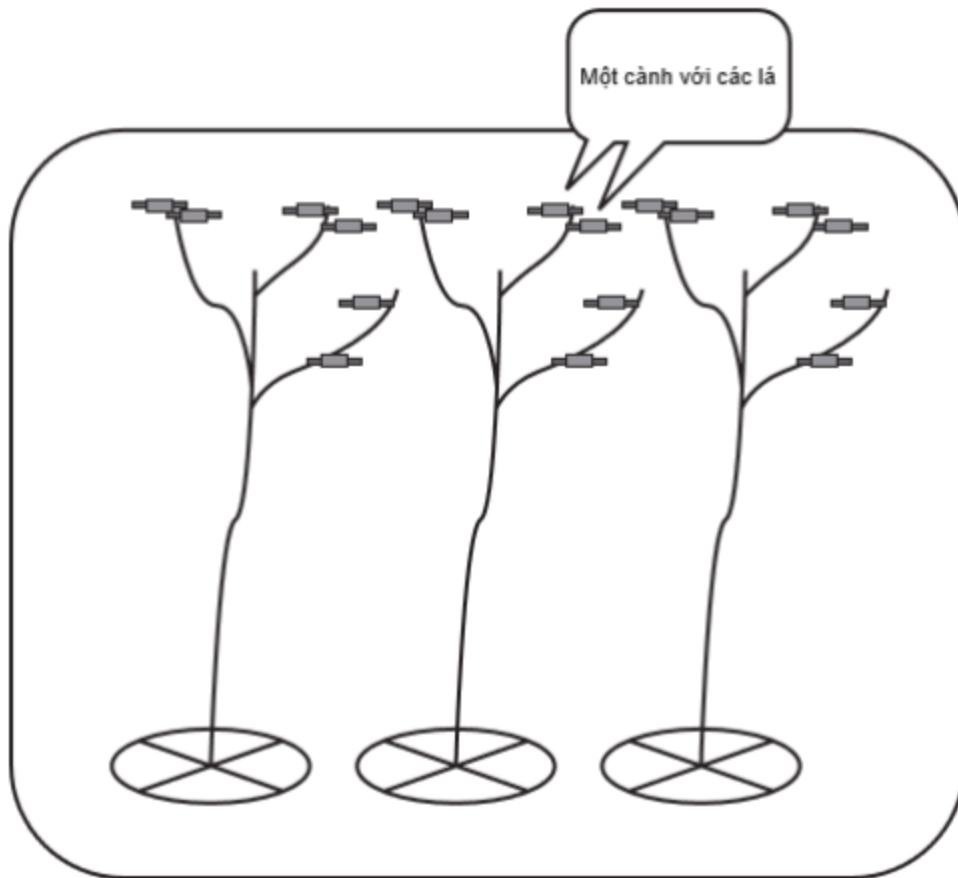


Ví dụ minh họa quá trình trực quan để tìm kiếm yêu cầu



Cây quản lý người dùng

Kỹ thuật cây và rừng

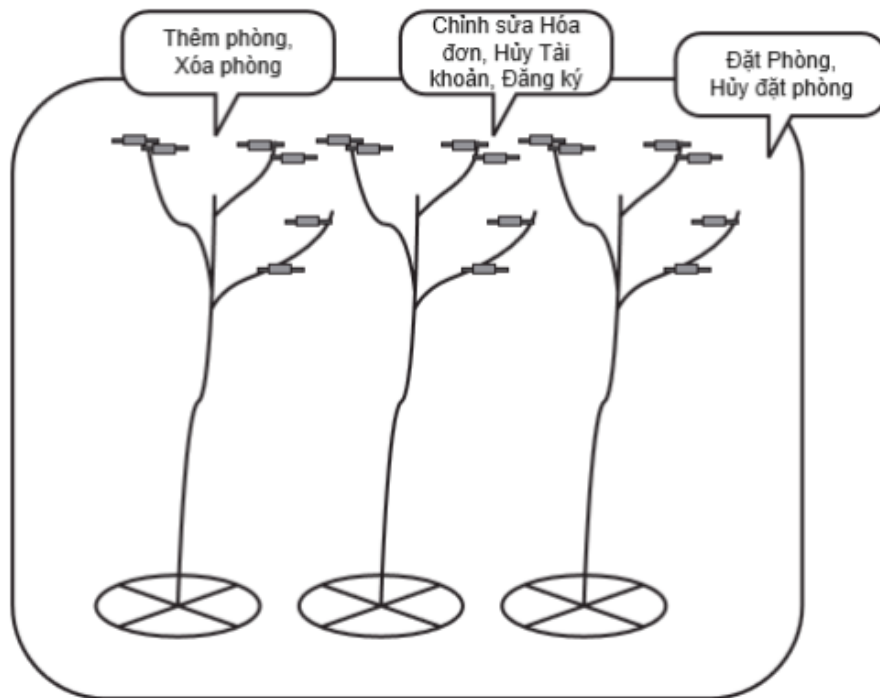


Quy Tắc CutFit

Có nhiều cách để xác định mục tiêu, một trong số đó là quy tắc CUTFIT

1. **Nhất quán:** yêu cầu không mâu thuẫn với yêu cầu khác
2. **Không mơ hồ:** yêu cầu được diễn giải theo một cách hiểu duy nhất
3. **Khả thi:** phải triển khai được từng yêu cầu trong điều kiện nguồn lực của môi trường hệ thống
4. **Độc lập:** không có **user story** này phụ thuộc vào **user story** khác
5. **Theo dõi được:** bạn phải có khả năng liên kết từng yêu cầu đến với người dùng và mục đích của họ

Ví dụ minh họa quá trình trực quan để tìm kiếm về cầu



Phần mềm đặt phòng và cây mô phỏng chức năng phần mềm

Ví dụ minh họa quá trình trực quan để tìm kiếm yêu cầu



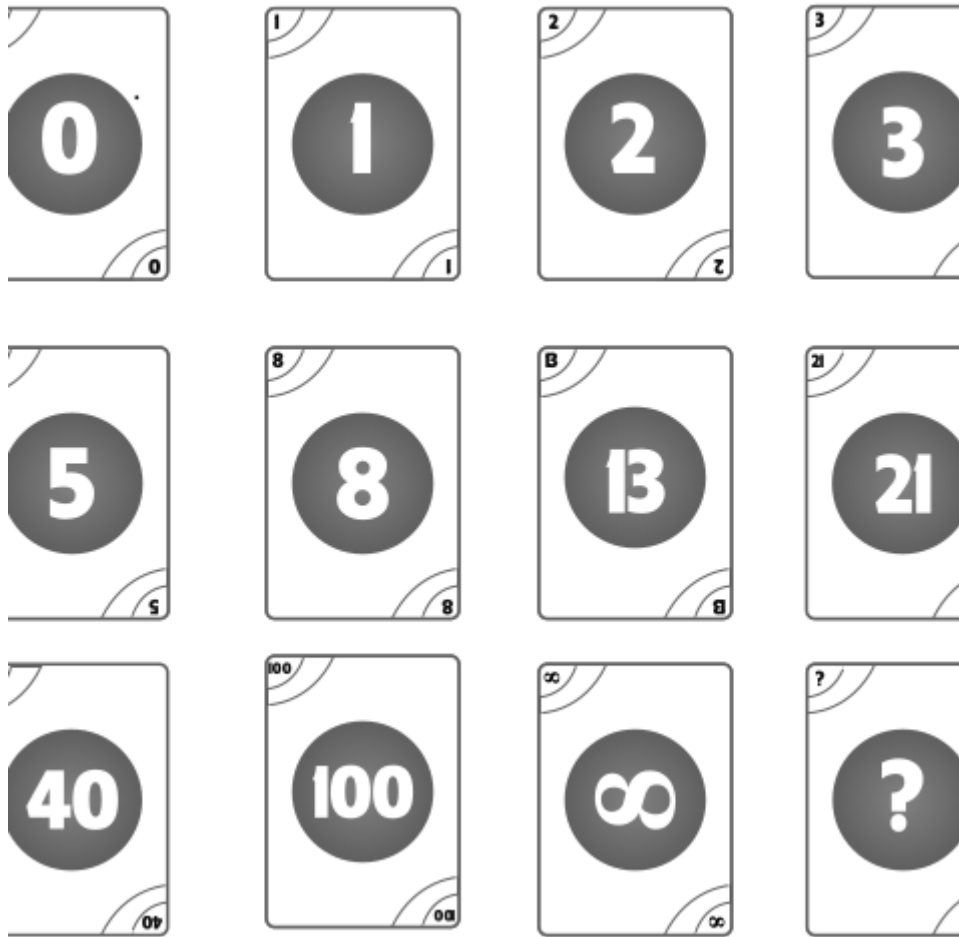
Cách tiếp cận cây và rừng đối với phát triển sản phẩm phần mềm

Ước lượng yêu cầu

1. Ước lượng yêu cầu là một trong các việc khó nhất của dự án phần mềm
2. Một vài con số thống kê:
 - a. 2/3 số dự án vượt quá đánh giá ước lượng ban đầu
 - b. 64% tính năng của phần mềm ít hoặc không bao giờ được sử dụng
 - c. Tính trung bình, dự án quá deadline 100%
3. 2 cấp độ đánh giá: đánh giá version và đánh giá lặp
4. Tập trung vào chức năng, không phải hành động
5. Tất cả thành viên của nhóm tham gia vào đánh giá

Bộ bài lập kế hoạch - PLANNING POKER

- The best way I've found for agile teams to estimate is by playing planning poker (Grenning 2002)



- Phương pháp này dựa trên tổ chức các cuộc họp ngắn và hiệu quả

Planning Poker là một kỹ thuật rất hiệu quả được sử dụng phổ biến để thực hiện ước tính trong Agile. Planning Poker kết hợp cả ba cách thức ước tính là dựa trên ý kiến chuyên gia, so sánh tương đối và chia nhỏ các hạng mục. Việc ước tính sử dụng Planning Poker khá nhanh chóng và mang lại kết quả đáng tin cậy. Nhóm **Scrum** có thể sử dụng **Planning Poker** để ước tính các hạng mục **Product Backlog (Agile Estimation – Ước tính linh hoạt)** hoặc các hạng mục công việc trong Sprint Backlog.

Để thực hiện kỹ thuật này, cần phải chuẩn bị các bộ bài **Planning Poker** đủ cho tất cả các thành viên tham dự. Bộ bài **Planning Poker** có nhiều kiểu khác nhau, tuy nhiên thông thường thì một bộ poker này chứa các quân bài thuộc dãy số: **0, 1, 2, 3, 5, 8, 13, 20, 40, 100**. Đây chính là dãy số **Fibonacci** đã được điều chỉnh một chút để phù hợp với việc ước tính. Dãy số này thể hiện các giá trị ước tính, không phụ thuộc vào đơn vị ước tính. Do đó, kỹ thuật này có thể sử dụng để ước tính với các đơn vị khác nhau, chẳng hạn là điểm tương đối hoặc giờ lý tưởng. Các nhóm có thể tự thiết kế bài cho mình hoặc là mua các loại có sẵn trên thị trường.

Chuẩn bị cho cuộc họp

- Chuyên gia về yêu cầu phải hiểu rõ về các user story
- Chuẩn bị cho mỗi thành viên một tập lá bài
- Mỗi lá bài thể hiện một ước lượng đánh giá cho một story



Chuẩn bị cho cuộc họp

- Ví dụ về giá trị đánh giá cho các lá bài:
 - 0, 1, 2, 3, 5, 8, 13, 20, 40, 100
 - 1, 2, 3, 5, 8, 13, BIG
 - $\frac{1}{2}$, 1, 2, 3, 4, 5, 6, 7, ∞

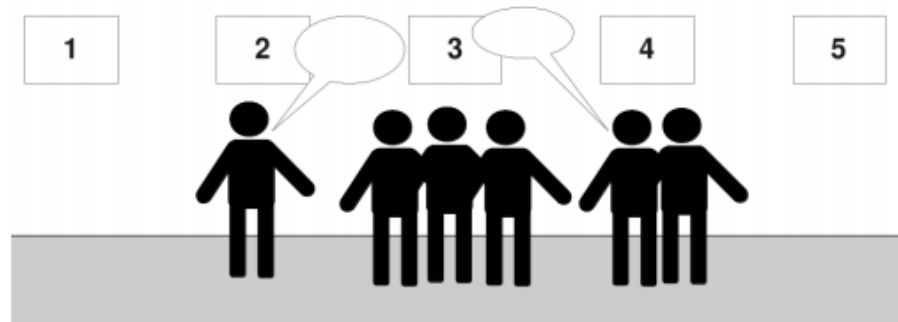


Cuộc họp

- Mỗi thành viên cầm bộ bài
- Người trung gian trình bày về user story trong thời gian không quá 2 phút
- Sau đó các thành viên hỏi thông tin về user story
- Mỗi thành viên chọn một thẻ bài mà không tiết lộ cho ai biết
- Khi tất cả thành viên chọn xong, tất cả lá bài được lật cùng lúc
- Trong vòng đầu tiên, có thể sự ước lượng của các thành viên khác nhau đáng kể

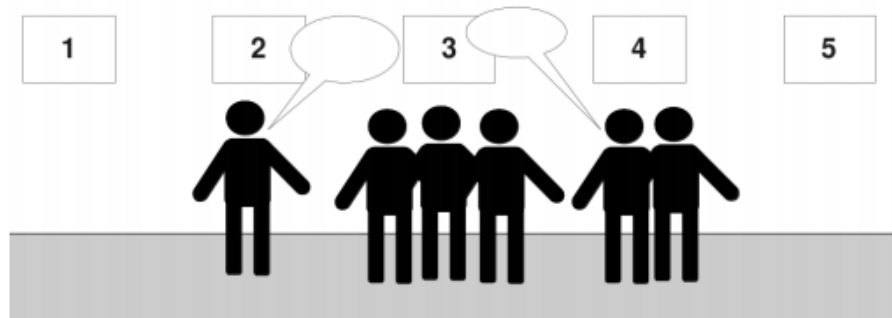
Cuộc họp

- Trong trường hợp có sự khác biệt lớn, giá trị thấp nhất và cao nhất của lá bài có thể chứa thông tin giải thích
- Sau đó các thành viên thảo luận ngắn về user story và ước lượng đánh giá của các thành viên
- Các thành viên tiếp tục suy nghĩ độc lập về ước lượng mới và sau đó cùng lúc công bố lá bài
- Nếu ước lượng vẫn khác nhau nhiều, tiến trình này tiếp tục được lặp lại



Cuộc họp

- Khi ước lượng gần giống nhau, kết thúc và chuyển sang user story tiếp theo
- Nếu trong lần thứ 3, các ước lượng vẫn khác nhau nhiều, có 3 khả năng tiếp theo:
 - Cùng nhau thử tiếp lần nữa
 - Người dùng sẽ chia story ra thành các phần nhỏ hơn
 - Lấy giá trị cao nhất, thấp nhất hoặc trung bình



Hãy nhớ, các con số đó chỉ là ƯỚC LƯỢNG mà thôi.

Cách ước lượng này giúp cho mọi thành viên trong nhóm đưa ra ý kiến một cách độc lập và từ đó họ sẽ giúp nhau tìm ra một cách hiểu đúng về hạng mục sẽ phải làm.

Ví dụ

- User story: Tạo một yêu cầu bán hàng
- Đội gồm 7 thành viên
- Vòng 1:



Ví dụ



- Thành viên thứ 3 và thứ 6 giải thích lý do ước lượng story
- Vòng 2:



Ví dụ



- Vòng bỏ phiếu tiếp theo có thể xảy ra
- Có thể lấy giá 3 hoặc 5 cho ước lượng story

Ưu điểm của phương pháp lá bài lập kế hoạch

- Thu được nhiều ý kiến đóng góp của các chuyên gia
- Hội thoại giữa các thành viên đưa đến kết quả ước lượng chính xác hơn
- Các nghiên cứu chỉ ra ước lượng trung bình và thảo luận nhóm sẽ cho kết quả tốt hơn

Planning Poker



Khuyết điểm của phương pháp lá bài lập kế hoạch

- Khó thu xếp cuộc họp của tất cả thành viên
- Người điều phối phải cẩn thận và điều khiển cuộc họp diễn ra đủ ngắn
- Một số nhân tố có thể ảnh hưởng đến ước lượng: chính trị, văn hóa, chủ nghĩa cá nhân

Planning Poker



Phần mềm bộ bài kế hoạch

- Vào trang web planningpoker.com
- Các thành viên đăng ký thành viên và tiến hành đánh giá User Story



Có thể tham khảo thêm ở video sau:

<https://www.youtube.com/watch?v=NII6mvGGkvg>

***Quản lý mục tiêu, ước
lượng Story***

Mục tiêu của Sprint

- Là một tập các mục tiêu cần đạt trong một Sprint sau khi triển khai một phần của Product Backlog
- Là một bộ các yêu cầu gắn kết khiến nhóm phát triển làm việc cùng nhau thay vì phân rã mỗi người một việc
- Nếu công việc phức tạp hơn dự kiến, nhóm sẽ cộng tác với Product Owner để thương lượng lại phạm vi của Sprint Backlog trong Sprint



Lợi ích khi xác định mục tiêu của Sprint

- Hỗ trợ xác định độ ưu tiên
 - Đầu tiên xác định mục tiêu
 - Sau đó chọn các story liên quan đến mục tiêu
- Tạo sự tập trung và hỗ trợ nhóm tốt hơn từ khách hàng
- Thuận tiện khi phân tích phản hồi
- Hỗ trợ giao tiếp giữa các bên liên quan
- Nhận thông tin phản hồi



Loại tương tác

- Nếu Story cần tương tác với con người, sẽ có giá trị 3
- Nếu Story chỉ cần tương tác với ứng dụng khác, theo một giao thức được định nghĩa rõ ràng, sẽ có giá trị 1

Loại tương tác	Mô tả	Giá trị
Đơn giản	Giao diện được định nghĩa rõ ràng	1
Trung bình	Giao diện động	2
Phức tạp	Tương tác với con người	3

Quy tắc nghiệp vụ

- Tính toán độ phức tạp dựa trên quy tắc nghiệp vụ
 - Nếu story chỉ có một quy tắc nghiệp vụ, sẽ có giá trị 1
 - Nếu có từ 1 đến 3 quy tắc nghiệp vụ, sẽ có giá trị 2
 - Nếu có nhiều hơn 3 quy tắc nghiệp vụ, sẽ có giá trị là 3

Quy tắc nghiệp vụ	Mô tả	Giá trị
Đơn giản	1 quy tắc	1
Trung bình	1-3 quy tắc	2
Phức tạp	>3 quy tắc	3

Số lượng thực thể liên quan đến story

- Nếu chỉ có một thực thể, story nên có giá trị là 1
- Nếu có từ 1 đến 3 thực thể, story nên có giá trị là 2
- Nếu có lớn hơn 3 thực thể, story nên có giá trị là 3

Thực thể	Mô tả	Giá trị
Đơn giản	1 Thực thể	1
Trung bình	1-3 Thực thể	2
Phức tạp	>3 Thực thể	3

Thao tác dữ liệu (CRUD)

Loại thao tác dữ liệu	Mô tả	Giá trị
Đơn giản	Đọc, Xóa	1
Trung bình	Tạo	2
Phức tạp	Cập nhật	3

Ví dụ

- Ví dụ ước lượng cho story **“thêm phòng họp”**
- Tính toán điểm chưa điều chỉnh UP trước khi xem xét đến yếu tố môi trường ở phần sau
 - Kiểu tương tác: 3 điểm
 - Quy tắc nghiệp vụ: 2 điểm
 - Số lượng thực thể: 1 điểm
 - Thêm mới, xóa, cập nhật, xóa (CRUD): 2 điểm
 - UP = 7 điểm