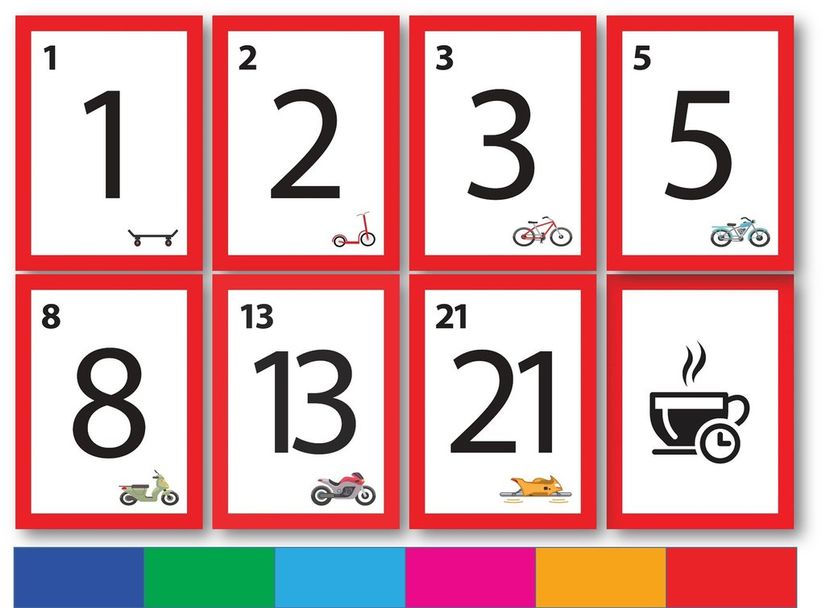
*Tìm hiểu trò chơi tính điểm cho story: Poker?*

*Viết: nhóm vai trò của từng người trong nhóm. Productbacklog, user story của dự án bạn?*

------------------------------------------------------------------

### Planning Poker

Phương pháp đầu tiên được gọi là "Planning Poker" và ban đầu được mô tả bởi James Grenning (2002) và được Mike Cohn phổ biến trong ‘Agile Estimating and Planning’ (2005).



Trò chơi diễn ra như sau:

* Bước 1: User Story (câu chuyện của người dùng) được thông báo cho toàn bộ nhóm, trong đó mỗi thành viên trong nhóm ước tính số điểm Story Points và không tiết lộ lựa chọn của mình cho người khác,
* Bước 2: Tất cả các ước tính được công bố cùng một lúc,
* Bước 3: Ước tính cao và thấp được giải thích bởi những người chọn và sau đó sẽ được đưa ra thảo luận
* Sau một hồi thảo luận, mỗi thành viên chọn một lá bài để ước tính lại và các lá bài của họ lại đồng thời được lật lên. Quy trình này sẽ lặp đi lặp lại cho tới khi đạt được sự đồng thuận của các thành viên hoặc họ quyết định rằng ước tính về tính năng này cần hoãn lại cho tới khi có thêm các thông tin bổ sung.

Một yếu tố quan trọng khác của Planning Poker là các giá trị cho phép của các ước tính chỉ có thể là một số trong số dãy Fibonacci: 1, 2, 3, 5, 8 hoặc một dãy tương tự nó như 1, 2, 4, 8. Điều này hạn chế việc phán đoán giá trị chính xác khó khăn hơn khi mọi thứ trở nên lớn hơn.

Quá trình này bỏ qua một yếu tố quan trọng. Khi nhóm nghiên cứu ước tính User Story đầu tiên, mỗi thành viên trong nhóm làm thế nào để biết điều gì tạo thành một Story Point? Không có point tham khảo để họ so sánh.

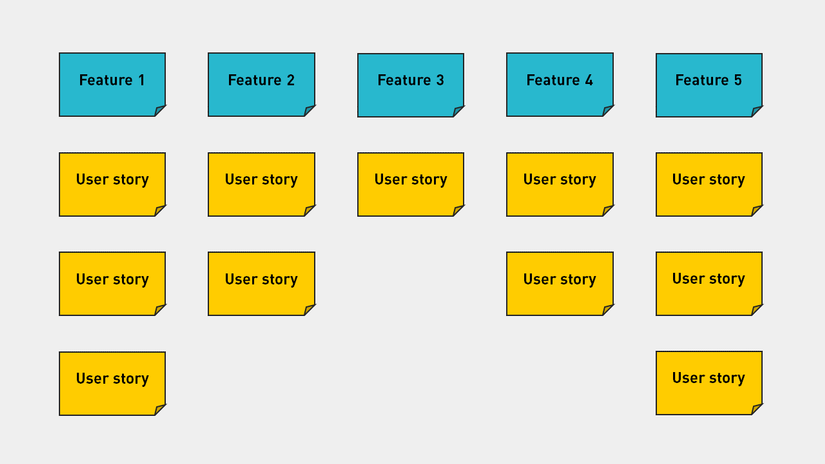
Hãy nhớ rằng, chúng tôi đang sử dụng các phương pháp so sánh, và với User Story đầu tiên, không có gì để so sánh cả. Cả nhóm tìm thấy User Story nhỏ nhất, và đồng ý gọi đó là 1 User Story. Tất cả các ước tính tiếp theo sẽ liên quan đến nó.

### Ước tính hai giai đoạn

Phương pháp ước tính này tránh được vấn đề không có point tham khảo bằng cách bắt đầu quá trình với User Story thứ hai, không phải là User Story đầu tiên, do đó có thể so sánh ngay lập tức.

Phương pháp này có hai giai đoạn.

**1, giai đoạn đầu tiên:**

user story ban đầu được đặt ở dạng đơn giản: trên tường hoặc bảng chẳng hạn.

* Thành viên đầu tiên của nhóm đặt user story của họ tại vị trí:
* Ở bên trái của user story đầu tiên, mục đích thể hiện nó đòi hỏi ít công việc hơn là user story đầu tiên,
* Ở bên phải, chỉ ra rằng nó đòi hỏi nhiều công việc hơn, hoặc
* Bên dưới một user story khác, chỉ ra rằng nó đòi hỏi cùng một lượng công việc.

Mỗi thành viên trong nhóm thay phiên nhau đặt một user story mới xung quanh những user story mà người khác đã đặt.

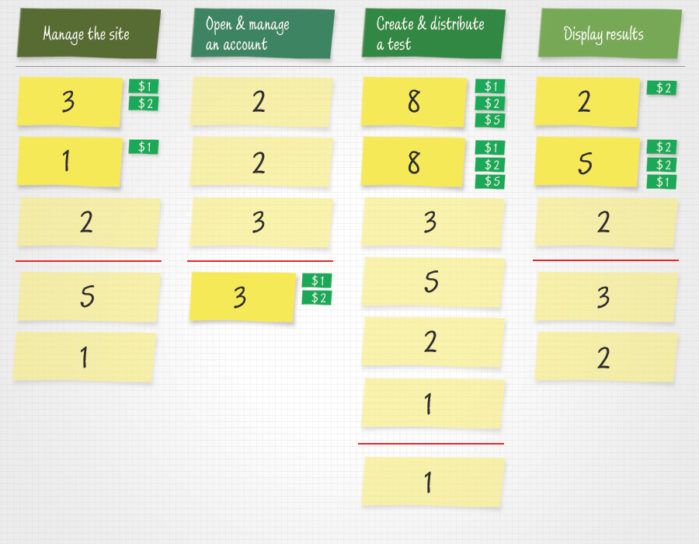
Các thành viên trong nhóm có thể có tùy chọn khác trong lượt của họ:

* họ có thể di chuyển một user story được đặt trước đó đến một vị trí khác nếu họ không đồng ý với vị trí ban đầu.  - Tiếp tục chơi cho đến khi hết user story và không ai muốn sắp xếp lại thứ tự nữa.

Tại thời điểm này, các user story có một thứ tự tương đối, nhưng vẫn chưa được gán user point. Điều này được thực hiện ở giai đoạn hai.

**2,Giai đoạn hai:**

* Cũng đòi hỏi một chuỗi Fibonacci, hoặc một cái gì đó tương tự.
* Số 1 được đặt phía trên cột câu chuyện ngoài cùng bên trái, đại diện cho user story nhỏ nhất.
* Thành viên nhóm đầu tiên lấy số tiếp theo: 2 và đặt nó lên trên những user story anh / cô ấy đánh giá là gấp đôi công việc của cột đầu tiên.
* Người chơi tiếp theo nhận số tiếp theo và gán nó vào một cột user story tương tự.
* Mỗi người chơi có một tùy chọn khác nhau, để thay thế số trước đó bằng số của họ, ví dụ: họ có thể cảm thấy rằng những user story được đánh số 5 có thể trong thực tế là 8
* Tùy chọn, các quy tắc có thể cho phép người chơi cũng di chuyển một user story của người sau khi nó được gán một số được đánh giá rõ ràng là vị trí sai.
* Khi tất cả các số đều đã được đặt và mọi người đều đồng ý, các user story được làm tròn lên.



Trong cả hai lượt chơi, giá trị 0 cũng có thể được sử dụng trong trường hợp các user story nhỏ nhất thực sự được coi là nhiệm vụ 30 phút và có thể được gộp vào những công việc khác với chi phí ít hoặc không có chi phí.

Lưu ý rằng chỉ có một số lượng giới hạn các user story bằng 0 được miễn phí.

Các phương pháp ước tính này có thể được sử dụng trước khi một dự án được xoay vòng và khi user story mới được xác định hoặc những user story hiện có thay đổi.

Câu hỏi tiếp theo là, làm cách nào để biết có bao nhiêu story points có thể gán cho bất kỳ lần lặp nào? Đó chính là Velocity: biểu đồ thể hiện kết quả mà team đã làm được trong 1 sprint.

### Velocity

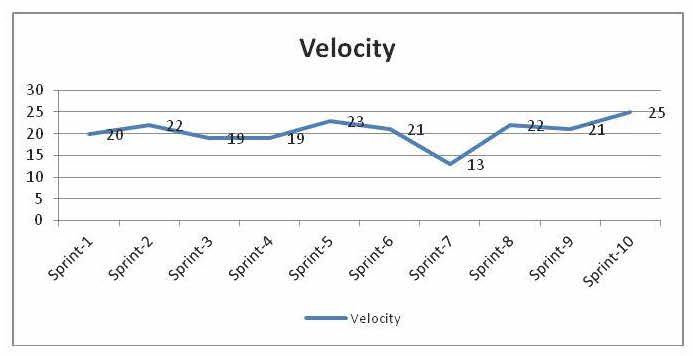
Vận tốc là thước đo số điểm mà một đội có thể thực hiện trong một lần lặp dựa trên hiệu suất trước đó. Điều quan trọng là các story points và vận tốc vẫn còn trừu tượng để tránh rơi vào thói quen cũ là ước lượng tuyệt đối dựa trên thời gian. Bằng suy luận, vận tốc sẽ gán một giá trị thời gian tuyệt đối cho một story points theo cách sau:

số người làm của nhóm cho mỗi story points = (số ngày có sẵn cho toàn bộ nhóm trong một lần lặp lại-các nhiệm vụ ( như kỳ nghỉ, cuộc họp, v.v.))/vận tốc.

Ví dụ: Một nhóm gồm 5 người có 50 ngày làm việc nhóm cho sprint 2 tuần. 1 người thì trung bình 1 tuần mất 1 ngày cho mấy việc linh tinh như meeting, nghỉ ngơi, và các vấn đề khác. vậy cả khâu 2 tuần mỗi người sẽ mất 2 ngày không làm việc. Vận tốc của nhóm cố định là 46,5. Vậy số người thực tế cần dùng là: 50 - (5 x 2) / 46,5 = 0,86

Điều quan trọng là nhận ra rằng giá trị 'ngày làm việc của team cho 1 story point' là trung bình của tất cả các thành viên trong nhóm và tất cả các thành viên trong nhóm đều phải tham gia vào vòng lặp thì vận tốc mới có ý nghĩa.  Nếu 3 lập trình viên có kinh nghiệm được thay thế bằng các thành viên mới trong một sprint thì vận tốc cho sprint đó phải được đánh giá lại.

Mặc dù ước tính thời gian story point có thể dễ dàng nhưng nó lại không dễ làm.



### Tổng hợp

Một việc cần được ghi nhớ trong suốt quy trình estimate này là ước lượng không phải là đã kết thúc công việc. Theo triết học Agile, cần xác định được việc nỗ lực ít nhất có thể mà vẫn hoàn thành được công việc. Các ước tính là cần thiết để ưu tiên cho việc xử lý những vấn đề còn tồn đọng và để đạt được sự phân bổ có ý nghĩa hơn cho mỗi sprint. Bằng cách làm như vậy, user story bắt đầu được thực hiện, không phải chỉ là ước tính và lập kế hoạch.

Cuối cùng, các ước tính tốt nhất là từ các quan điểm tập thể của toàn bộ nhóm. Việc sử dụng ý kiến của người quản lý hoặc chuyên gia sẽ giúp các ước tính lạc quan hơn. Ngược lại, việc sử dụng phần lớn ý kiến của các cá nhân mới vào nhóm có thể dẫn đến ước tính khối lượng thời gian và công việc lớn hơn mức cần thiết. Sau tất cả Agile là bao gồm tất cả mọi người.

Câu 2:

Tạo ra ứng dụng cho những người có nhu cầu về công nghệ, để họ có thể theo dõi các sản phẩm mà họ đang quan tâm, đồng thời khám phá những sản phẩm mới bằng cách theo dõi, các thành viên chia sẽ đánh giá của mình về những sản phẩm của các cửa hàng mà họ đã tin dùng.

1. Đăng ký, đăng nhập, đăng xuất, xem hồ sơ. Chỉnh sửa hồ sơ, xóa hồ sơ, duyệt người dùng.
2. Thêm, xóa, sửa, duyệt sản phẩm.
3. Thêm, sửa, duyệt, xem cửa hàng
4. Thêm, sửa, xóa đánh giá
5. Quản lý người dùng
6. Quản lý sản phẩm(đt)
7. Quản lý cửa hàng
8. Quản lý đánh giá

Phần mềm quản lý trang mạng bán đt iphone

Phân chia vai trò:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Name | Account | Roles | DateStart | Date End |
| 1 | Cao Phước Sơn | soncao | Product owner | 13/11/2020 |  |
| 2 | Nguyễn Văn Tỵ | tynguyen | Scrum master | 13/11/2020 |  |
| 3 | Nguyễn Văn Tỵ  Cao Phước Sơn | sonty | Team lead | 13/11/2020 |  |
| 4 | Cao Phước Sơn | soncao | Dev | 13/11/2020 |  |
| 5 | Nguyễn Văn Tỵ | tynguyen | Dev | 13/11/2020 |  |

Product backlog

|  |  |  |
| --- | --- | --- |
| Story ID | Story Name | Point 83 |
| Actor Guest | | |
| 1 | View home page | 4 |
| 2 | View product list | 4 |
| 3 | View product detail | 2 |
| 4 | View product Compare | 2 |
| 5 | Manage Shopping Cart (Add, increase, decrease, remove) | 4 |
| 6 | Registration | 4 |
| 7 | Chat with Sale | 4 |
| Actor: Customer | | |
| 8 | Login | 2 |
| 9 | Make orders | 2 |
| 10 | Order management (View, cancel request) | 6 |
| 11 | View Shipping Progress | 2 |
| 12 | Online Payment | 4 |
| 13 | Logout | 1 |
| Actor: Order Processing | | |
| 14 | View New Order | 2 |
| 15 | Order Management (View, Short, Update) | 4 |
| 16 | Shipping Management | 6 |
| 17 | Offline Payment | 2 |
| 18 | Payment Mornitoring | 2 |
| Actor: Sale Manager | | |
| 19 | Product Management | 6 |
| 20 | Store Management | 4 |
| 21 | Inventory Management | 6 |
| 22 | Price Management | 4 |
| 23 | Promotion Management | 4 |
| Actor: Administration | | |
| 24 | User Management | 6 |