

1. Bạn có một ứng dụng quản lý công việc và muốn triển khai tính năng "Tạo công việc mới". Hãy ước lượng mức độ phức tạp của User Story này bằng phương pháp Story Points.

Yêu cầu:

Sử dụng phương pháp Story Points để ước lượng độ phức tạp của User Story này.

Cung cấp lý do giải thích tại sao bạn chọn Story Points này.

Ví dụ về cách ước lượng với Story Points:

- a. User Story: "Là người dùng, tôi muốn tạo công việc mới để có thể theo dõi tiến độ công việc."
 - b. Ước lượng Story Points: 5 điểm Story (mức độ khó trung bình)
 - c. Lý do giải thích:
 - i. Việc tạo công việc mới có thể không quá phức tạp, nhưng sẽ yêu cầu một số bước như thiết kế form, xử lý dữ liệu và hiển thị thông báo. Vì vậy, mức độ khó có thể ở mức trung bình (5 điểm trong dãy Fibonacci).
- User Story: Tạo công việc mới
 - Vai trò: Người dùng
 - Mục đích: Là người dùng tôi muốn tạo công việc mới
 - Giải thích: Để có thể theo dõi tiến độ công việc
 - Ước lượng Story Point : 5
 - Các bước

B1: Từng thành viên trong nhóm điền 1 số làm story point

B2: Tổ chức Planning Poker xem các thành viên đã đánh giá điểm của User Story

+ Lấy trung bình (Cộng tất cả lấy trung bình)

+ Lấy kết quả Story Point xuất hiện nhiều nhất

- Có kết quả Story Point -> Đánh giá độ phức tạp theo cấp độ

Giải thích: Việc tạo công việc mới có thể không quá phức tạp, nhưng sẽ yêu cầu một số bước như thiết kế form, xử lý dữ liệu và hiển thị thông báo. Vì vậy, mức độ khó có thể ở mức trung bình

2. Bạn có một ứng dụng web bán hàng trực tuyến và muốn triển khai tính năng "Thanh toán đơn hàng". Hãy ước lượng mức độ phức tạp của User Story này bằng phương pháp Story Points.

Yêu cầu:

- Sử dụng phương pháp Story Points để ước lượng độ phức tạp của User Story này
 - Cung cấp lý do giải thích tại sao bạn chọn Story Points này.
-
- User story: Thanh toán đơn hàng
 - Vai trò: Khách hàng
 - Mục đích: Muốn thanh toán hóa đơn
 - Ước lượng Story Point: 13
 - Giải thích: Tính năng thanh toán đơn hàng khá phức tạp vì phải xử lý nhiều bước như xác nhận thông tin đơn hàng, tính tổng tiền, tích hợp cổng thanh toán, nhận kết quả trả về và cập nhật trạng thái đơn hàng
 - Chọn phương pháp ước lượng story point
- + Các thành viên trong team đánh giá 1 số trong dãy Fibonacci làm Story Point
- + Tính Trung bình điểm Story Point hoặc lấy kết quả nhiều người chọn nhất làm Story Point
- Có điểm Story Point -> Đánh giá độ phức tạp

3. [Bài tập] Planning Poker – Kỹ thuật ước lượng công việc

1. Mục tiêu

- Hiểu và thực hành sử dụng kỹ thuật Planning Poker trong ước lượng công việc.

2. Mô tả

- Sử dụng phương pháp Planning Poker, nhóm bạn sẽ ước lượng công việc cho User Story “Tạo tính năng giỏ hàng trong ứng dụng bán hàng trực tuyến”. Cả nhóm sẽ thảo luận và chọn điểm Story dựa trên độ phức tạp của công việc.

Yêu cầu:

- Mỗi thành viên chọn một lá bài với số điểm Story tương ứng với độ phức tạp của User Story.
- Sau khi mở bài, thảo luận và đi đến sự đồng thuận về mức độ phức tạp.
- Ghi lại kết quả ước lượng và giải thích lý do nếu có sự khác biệt.

Ví dụ về cách thực hiện:

- User Story: "Là người dùng, tôi muốn thêm sản phẩm vào giỏ hàng để có thể mua hàng trong một đơn hàng duy nhất."
- Điểm Story ước lượng (Planning Poker):
 - Mỗi thành viên trong nhóm chọn một số điểm Story từ dãy Fibonacci (1, 2, 3, 5, 8, 13, 20, 40, 100).
 - Sau khi tất cả thành viên đã chọn, mở bài và thảo luận về lý do sự khác biệt trong các ước lượng.
- Giải thích sự khác biệt: Nếu có sự khác biệt trong các điểm Story, nhóm sẽ thảo luận về lý do và đi đến sự đồng thuận.
- Story Point: Tạo giỏ hàng
- Vai trò: Khách hàng
- Mục đích: Tạo giỏ để thêm danh sách sản phẩm trong giỏ hàng
- Giải thích: Muốn xem danh sách sản phẩm trong giỏ hàng, có thể xem chi tiết thông tin từng sản phẩm
- Điểm Story Point ước lượng:

- + Mỗi thành viên trong nhóm chọn 1 điểm Story Point trong dãy số Fibonacci
- + Sau khi các thành viên trong nhóm chọn thì sẽ thảo luận để lấy kết quả Story Point cuối cùng tính điểm trung bình Story Point hoặc lấy kết quả nhiều người chọn nhất làm Story Point. Trong trường hợp có 2 or nhiều bằng điểm thì lấy Story Point có độ phức tạp cao hơn

4. Sử dụng phương pháp Planning Poker, nhóm bạn sẽ ước lượng công việc cho User Story “Tạo tính năng đăng nhập trong ứng dụng ngân hàng trực tuyến”. Cả nhóm sẽ thảo luận và chọn điểm Story dựa trên độ phức tạp của công việc.

Yêu cầu:

- a. Mỗi thành viên chọn một lá bài với số điểm Story tương ứng với độ phức tạp của User Story.
- b. Sau khi mở bài, thảo luận và đi đến sự đồng thuận về mức độ phức tạp.
- c. Ghi lại kết quả ước lượng và giải thích lý do nếu có sự khác biệt.

- User story: Tính năng đăng nhập
- Vai trò : Người dùng
- Mục đích: Đăng nhập để sử dụng hệ thống ngân hàng
- Sử dụng ước lượng (Planning Poker):
 - + Từng thành viên trong nhóm chọn điểm đánh giá Story Point theo dãy Fibonacci
 - + Thảo luận để lấy ra Story Point cuối cùng

Thành viên trong nhóm	Điểm Story Point
Nguyễn Văn A	5

Nguyễn Văn B	5
Nguyễn Văn C	8

5. Dựa trên một ứng dụng quản lý công việc, nhóm bạn sẽ sử dụng các phương pháp Story Points, Planning Poker, và Velocity để lựa chọn các User Stories cho Sprint tiếp theo và lên kế hoạch thực hiện.

Yêu cầu:

- Sử dụng các phương pháp ước lượng:
 - a. Story Points: Ước lượng mức độ phức tạp của các User Stories trong Product Backlog. Sử dụng dãy số (1, 2, 3, 5, 8, 13, 21) để ước lượng.
 - b. Planning Poker: Tổ chức cuộc họp Planning Poker để thảo luận và chọn điểm Story cho các User Stories.
- Xác định mục tiêu Sprint:
 - c. Đưa ra mục tiêu Sprint cụ thể cho nhóm, chẳng hạn như "Hoàn thành tính năng giỏ hàng và chức năng đăng nhập".
- Lập kế hoạch phân chia công việc:
 - d. Phân chia User Stories thành các tasks nhỏ hơn và ước lượng thời gian/công sức cho mỗi task.
 - e. Đảm bảo rằng các tasks được phân bổ hợp lý cho nhóm phát triển trong thời gian Sprint.
- Sử dụng Velocity:
 - f. Dựa trên Velocity từ Sprint trước, dự báo khả năng hoàn thành công việc trong Sprint tiếp theo. Velocity giúp nhóm xác định số lượng điểm Story có thể hoàn thành và lập kế hoạch cho Sprint tiếp theo.
- Ứng dụng quản lý công việc

1. Liệt kê các User Stories trong Product Backlog

1. US1 – Tạo công việc mới:
Là một người dùng, tôi muốn tạo công việc mới với tên và mô tả để quản lý các nhiệm vụ của mình.
2. US2 – Tìm kiếm công việc:
Là một người dùng, tôi muốn tìm kiếm công việc theo tên để nhanh chóng tìm đúng task.
3. US3 – Phân loại công việc:
Là một người dùng, tôi muốn lọc công việc theo trạng thái hoặc mức độ ưu tiên.

2. Ước lượng Story Points bằng phương pháp Planning Poker

User Story được chọn để ước lượng: US1 – Tạo công việc mới

Các thành viên đưa ra Story Points:

Thành viên	Điểm Story Point
Nguyễn Văn A	5
Nguyễn Văn B	8
Nguyễn Văn C	8

Thảo luận Planning Poker

- A cho rằng tính năng đơn giản → 5 points
- B & C nói có nhiều phần: giao diện + API + validate → 8 points
- Sau thảo luận, nhóm thống nhất: tính năng liên quan cả UI – backend – xử lý lỗi → phức tạp hơn mức trung bình.

Kết luận: Story Point cuối cùng cho User Story = 8

3. Xác định mục tiêu Sprint (Sprint Goal)

Sprint Goal:

“Hoàn thành tính năng tạo công việc mới, bao gồm giao diện, API thêm mới và hệ thống kiểm tra dữ liệu.”

4. Lập kế hoạch phân chia công việc (Task Breakdown)

User Story: Tạo công việc mới (8 Story Points)

Tên Task	Mô tả chi tiết	Ước lượng thời gian
Giao diện tạo công việc	Thiết kế giao diện cho phép nhập tên và mô tả công việc	8h
API thêm mới công việc	Viết API backend để nhận dữ liệu và lưu vào DB	4h
Validate dữ liệu	Hiển thị thông báo lỗi khi nhập sai hoặc thiếu dữ liệu	8h

5. Sử dụng Velocity để lập kế hoạch Sprint

Velocity từ Sprint trước: 20 Story Points

Velocity là số Story Points mà nhóm đã hoàn thành ở Sprint trước.

Tính toán lượng công việc có thể làm ở Sprint tiếp theo:

- Velocity = 20 SP
- User Story “Tạo công việc mới” = 8 SP
- Nhóm còn có thể nhận thêm các User Story khác:
- Tổng: $8 + 5 + 5 = 18 \text{ Story Points} \leq 20 \rightarrow$ phù hợp Velocity

6. Dựa trên một ứng dụng quản lý công việc, bạn sẽ thực hiện Sprint Planning cơ bản. Mục tiêu là xác định mục tiêu Sprint, chọn các User Stories từ Product Backlog và ước lượng công việc.

Yêu cầu:

1. Xác định Mục tiêu Sprint:
 - Lựa chọn một mục tiêu Sprint (ví dụ: "Hoàn thành tính năng quản lý công việc cho người dùng").
 - Mục tiêu này phải rõ ràng và có giá trị đối với người dùng hoặc khách hàng.
2. Chọn User Stories từ Product Backlog:
 - Chọn ít nhất 2 User Stories từ Product Backlog có liên quan đến mục tiêu Sprint.
 - Giải thích lý do tại sao chọn những User Stories này và cách chúng giúp đạt được mục tiêu Sprint.
3. Phân chia User Stories thành các task nhỏ:
 - Phân chia mỗi User Story thành các task nhỏ hơn để dễ thực hiện trong Sprint.
 - Ví dụ: Nếu một User Story là "Tạo tính năng đăng ký tài khoản", các task có thể là "Tạo form đăng ký", "Xử lý API đăng ký", "Hiển thị thông báo đăng ký".
4. Ước lượng công việc cho mỗi User Story:
 - Ước lượng công việc cho mỗi User Story theo phương pháp T-shirt Sizes (Small, Medium, Large).
 - Cung cấp lý do giải thích tại sao bạn chọn kích cỡ này.

User Story	Task	T-shirt Size	Giải thích size
Tạo công việc mới	-Thiết kế form tạo công việc -Xây dựng API lưu công việc vào cơ sở dữ liệu	Medium	-Độ phức tạp vừa phải khi mỗi task riêng lẻ không quá khó

	-Hiển thị thông báo khi tạo thành công/thất bại -Kiểm thử tính năng công việc		-Để hoàn thành thì cần ít nhất 2-3 ngày nên không phải small
Chỉnh sửa và xóa công việc	-Thiết kế giao diện danh sách công việc với nút chỉnh sửa và xóa -Xây dựng API cho chỉnh sửa công việc -Xây dựng API cho xóa công việc -Hiển thị thông báo khi chỉnh sửa hoặc xóa thành công/thất bại -Kiểm thử tính năng chỉnh sửa và xóa	Medium	Giống với user story 1 thì chỉ cần kết hợp backend và frontend nên không quá phức tạp

7. Trong Scrum, việc chia nhỏ User Stories thành các task cụ thể là rất quan trọng để nhóm có thể dễ dàng theo dõi và thực hiện công việc trong Sprint. Bài tập này yêu cầu bạn thực hành chia nhỏ một User Story thành các task nhỏ và hợp lý.

User Story: “Là một người dùng, tôi muốn đăng ký tài khoản để có thể sử dụng ứng dụng.”

Yêu cầu:

- Chia nhỏ User Story thành các task:
 - Dựa trên User Story trên, chia thành các task cụ thể.

- Đảm bảo mỗi task có thể hoàn thành trong một Sprint và không quá phức tạp.
 - Sử dụng các kỹ thuật chia nhỏ User Stories thành task (Dựa vào các bước phát triển, hành động người dùng, tính năng kỹ thuật).
2. Lý do lựa chọn các task:
- Cung cấp lý do tại sao bạn chọn các task này và cách chúng đảm bảo hoàn thành User Story một cách hiệu quả.
3. Ước lượng các task:
- Ước lượng độ phức tạp của mỗi task (ví dụ: sử dụng Story Points, T-shirt Sizes hoặc Ideal Days).

Chia nhỏ các task:

Task	Mô tả	Lý do	T-Shirt Size
Thiết kế giao diện form đăng ký	Tạo form đăng ký gồm các trường:tên,email,mật khẩu,xác nhận mật khẩu	Đây là bước cơ bản để người dùng nhập dữ liệu	Small
Validate dữ liệu	Kiểm tra các dữ liệu nhập(email,mật khẩu,xác nhận mật khẩu)	Kiểm tra dữ liệu người dùng và ngăn lỗi trước khi đưa lên server	Small
Xây dựng API backend	Lưu dữ liệu người dùng vào cơ sở dữ liệu	Lưu dữ liệu	Medium
Kết nối frontend với API	Gửi dữ liệu tới backend và xử lý phản hồi	Đảm bảo dữ liệu chính xác và an toàn	Medium
Hiển thị thông báo cho người dùng	Thông báo thành công hoặc thất bại	Giúp người dùng biết kết quả đăng ký	Medium
Kiểm thử tính năng đăng ký	Thực hiện kiểm tra các luồng	Đảm bảo User Story hoạt động đúng	Small

- Lý do lựa chọn các task
 - o Các Task được chia dựa theo các bước phát triển :Thiết kế giao diện,xử lý dữ liệu,kết nối api,thông báo và kiểm thử
 - o Mỗi task có thể hoàn thành trong vòng 1 - 2 ngày và không quá phức tạp
- Khi hoàn thành tất cả thì User Story được hoàn thành đầy đủ

8. Trong Scrum, việc chia nhỏ User Stories thành các task cụ thể là rất quan trọng để nhóm có thể dễ dàng theo dõi và thực hiện công việc trong Sprint. Bài tập này yêu cầu bạn thực hành chia nhỏ một User Story thành các task nhỏ và hợp lý.

User Story: “Là một người dùng, tôi muốn đăng ký tài khoản để có thể sử dụng ứng dụng.”

Yêu cầu:

1. Chia nhỏ User Story thành các task:
 - o Dựa trên User Story trên, chia thành các task cụ thể.
 - o Đảm bảo mỗi task có thể hoàn thành trong một Sprint và không quá phức tạp.
 - o Sử dụng các kỹ thuật chia nhỏ User Stories thành task (Dựa vào các bước phát triển, hành động người dùng, tính năng kỹ thuật).
2. Lý do lựa chọn các task:
 - o Cung cấp lý do tại sao bạn chọn các task này và cách chúng đảm bảo hoàn thành User Story một cách hiệu quả.
3. Ước lượng các task:
 - o Ước lượng độ phức tạp của mỗi task (ví dụ: sử dụng Story Points, T-shirt Sizes hoặc Ideal Days).

-Tính velocity của nhóm

Velocity sp1: $5+8=13$

Velocity sp2: $13+5=18$

-Tính velocity trung bình

$(13+18)/2=16$

- Sử dụng Velocity để lập kế hoạch cho Sprint tiếp theo

Nhóm có thể hoàn thành khoảng 16 Story Points trong Sprint tiếp theo.

Khi chọn User Stories cho Sprint, tổng Story Points không nên vượt quá 16.