

Bài 1: ứng dụng quản lý công việc

User story: tạo công việc mới

Vai trò : người dùng

Mục đích: Người dùng muốn thêm công việc mới

Giải thích mục đích: ghi chú lại những công việc và theo dõi, quản lý tiến độ công việc.

Dùng phương pháp User point

Tổ chức hợp Planning Poker

Tên thành viên	Điểm đánh giá story point
Nguyễn thanh tùng	5
Trần thị khánh huyền	3
Lê trung đông	5
Hải đăng	8

Điểm story Point = $(5+3+5+8)/4 = 5.25$

Kết luận: so sánh điểm Story Point với các Task trước để đưa ra kết luận

Việc tạo task không quá phức tạp, gồm thiết kế form, xử lý dữ liệu và thông báo

=> **Mức độ trung bình**

Bài 2: Ứng dụng bán hàng trực tuyến

User story: Thanh toán đơn hàng

Vai trò: Người dùng

Mục đích: Người dùng muốn thanh toán để hoàn tất quá trình mua hàng

Giải thích mục đích: Giúp khách hàng thực hiện giao dịch, xác nhận đơn hàng và nhận thông báo kết quả thanh toán.

Phương pháp: Story Points (Planning Poker)

Tên thành viên	Điểm đánh giá Story Point
Nguyễn Thanh Tùng	8
Trần Thị Khánh Huyền	5
Lê Trung Đông	8
Hải Đăng	13

Tính điểm trung bình:

$$(8 + 5 + 8 + 13)/4 = 8.5$$

Kết luận:

- So với User Story “Tạo công việc mới” (5.25 điểm), tính năng **Thanh toán đơn hàng** phức tạp hơn.
- Lý do: cần tích hợp cổng thanh toán, xử lý giỏ hàng, tính phí, bảo mật dữ liệu và kiểm thử nhiều trường hợp.
- Do đó, mức độ phức tạp **khá cao** (trên trung bình).

Bài 3:

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).

Tên thành viên	Điểm đánh giá Story Point
Nguyễn Thanh Tùng	5

Trần Thị Khánh Huyền	8
Lê Trung Đông	8
Hải Đăng	13

Tính điểm trung bình:

$$(5 + 8 + 8 + 13)/4 = 8.5$$

Giải thích sự khác biệt

- Thành viên chọn **5 điểm**: cho rằng chỉ cần form thêm sản phẩm và lưu dữ liệu cơ bản.
- Thành viên chọn **13 điểm**: đánh giá cao vì tính năng giỏ hàng liên quan đến nhiều thao tác (thêm/xóa sản phẩm, cập nhật số lượng, tính tổng tiền, đồng bộ dữ liệu).
- Sau thảo luận, nhóm thống nhất mức trung bình **8 điểm Story** để phản ánh độ phức tạp vừa phải nhưng có nhiều yếu tố cần kiểm thử.

3. Kết luận

- So với các User Story đơn giản hơn (ví dụ: “Tạo công việc mới” ~ 5 điểm), tính năng giỏ hàng phức tạp hơn.
- Lý do: có nhiều thao tác nghiệp vụ và liên quan đến dữ liệu động.
- Mức độ phức tạp: **khá cao (trên trung bình)**.

Bài 4:

User Story: “Là người dùng, tôi muốn đăng nhập vào ứng dụng ngân hàng trực tuyến để có thể truy cập tài khoản và thực hiện giao dịch.”

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).

Tên thành viên	Điểm đánh giá Story Point
Nguyễn Thanh Tùng	8
Trần Thị Khánh Huyền	5

Lê Trung Đông	8
Hải Đăng	13

Tính điểm trung bình:

$$(8 + 5 + 8 + 13)/4 = 8.5$$

Giải thích sự khác biệt

- Thành viên chọn **5 điểm**: cho rằng chỉ cần form đăng nhập và kiểm tra thông tin người dùng.
- Thành viên chọn **13 điểm**: đánh giá cao vì tính năng đăng nhập trong ứng dụng ngân hàng phải đảm bảo **bảo mật cao**, có thể bao gồm xác thực hai lớp (OTP, SMS, email), mã hóa dữ liệu, xử lý nhiều trường hợp lỗi.
- Sau thảo luận, nhóm thống nhất mức trung bình **8 điểm Story** để phản ánh độ phức tạp vừa phải nhưng có nhiều yếu tố quan trọng liên quan đến bảo mật và tích hợp hệ thống.

3. Kết luận

- So với các User Story cơ bản (ví dụ: “Tạo công việc mới” ~ 5 điểm), tính năng **Đăng nhập ngân hàng trực tuyến** phức tạp hơn.
- Lý do: liên quan đến bảo mật, xác thực đa lớp và xử lý nhiều tình huống nghiệp vụ.
- Mức độ phức tạp: **khá cao (trên trung bình)**.

Bài 5:

1. Sử dụng các phương pháp ước lượng

- User Story: Tạo công việc**
 - Vai trò**: Người dùng
 - Mục đích**: Người dùng muốn thêm công việc mới để quản lý và theo dõi tiến độ.
 - Giải thích lý do**: Giúp người dùng ghi chú lại các công việc, dễ dàng quản lý và kiểm soát tiến độ.

Planning Poker – Đánh giá Story Point

Tên thành viên	Điểm đánh giá
Dev A	5
Dev B	3
Dev C	8

Tính điểm trung bình:

$$(5 + 3 + 8)/3 = 5.3$$

Đánh giá mức độ:

- Mức độ trung bình, gồm các thao tác cơ bản như thiết kế form, xử lý dữ liệu nhập và hiển thị thông báo.
- User Story: Tìm công việc**
 - Vai trò:** Người dùng
 - Mục đích:** Người dùng muốn tìm kiếm công việc đã tạo để dễ quản lý.
 - Giải thích lý do:** Giúp người dùng nhanh chóng tra cứu, lọc và theo dõi công việc trong hệ thống.

2. Xác định mục tiêu Sprint Trong Sprint có 2 mục tiêu chính:

- Hoàn thành tính năng **Tạo công việc**.
- Hoàn thành tính năng **Tìm công việc**.

3. Lập kế hoạch phân chia công việc

Tên User Story	Phân chia Task	Người thực hiện	Ước lượng
Tạo công việc	Tạo giao diện	Dev A	5 SP
	Kiểm tra dữ liệu nhập	Dev A	3 SP
	Viết API để tạo công việc	Dev B	8 SP

Tổng điểm Story cho User Story “Tạo công việc” = 16 SP

4. Sử dụng Velocity

- Velocity Sprint trước: giả sử là **15 SP**.

- Sprint này: tổng điểm Story cho “Tạo công việc” = **16 SP**.
- Nhóm so sánh với Velocity trước để đánh giá khả năng hoàn thành.
- Kết luận: Nhóm có thể hoàn thành khối lượng công việc trong Sprint, nhưng cần điều chỉnh và phân bổ hợp lý để đảm bảo tiến độ.

Bài 6:

User Stories được chọn từ Product Backlog

1. Tạo công việc mới

- Lý do: Đây là chức năng cốt lõi, giúp người dùng thêm công việc để quản lý tiến độ.
- Giá trị: Cho phép người dùng bắt đầu sử dụng ứng dụng để ghi chú và theo dõi công việc.

2. Tìm kiếm công việc

- Lý do: Bổ sung cho chức năng tạo công việc, giúp người dùng nhanh chóng tra cứu và quản lý danh sách công việc.
- Giá trị: Tăng tính tiện lợi và hiệu quả khi sử dụng ứng dụng.

Phân chia User Stories thành các task nhỏ

User Story	Task	Người thực hiện	Ước lượng (T-shirt Size)	Lý do
Tạo công việc mới	Thiết kế giao diện form	Dev A	Small	UI cơ bản, ít phức tạp
	Kiểm tra dữ liệu nhập (validation)	Dev A	Medium	Cần xử lý nhiều trường hợp nhập sai
	Viết API lưu công việc	Dev B	Large	Liên quan backend, database, nhiều logic

	Hiển thị thông báo kết quả	Dev C	Small	Chỉ cần xử lý phản hồi từ API
Tìm kiếm công việc	Thiết kế giao diện tìm kiếm	Dev A	Medium	UI có filter, search bar
	Viết API tìm kiếm	Dev B	Large	Liên quan query dữ liệu, tối ưu hiệu năng
	Hiển thị kết quả tìm kiếm	Dev C	Medium	Cần xử lý danh sách, phân trang
	Kiểm thử chức năng tìm kiếm	Dev C	Small	Chủ yếu test nhiều trường hợp

Ước lượng công việc (T-shirt Sizes)

- **Tạo công việc mới:** Small + Medium + Large + Small → mức độ **khá phức tạp**.
- **Tìm kiếm công việc:** Medium + Large + Medium + Small → mức độ **khá phức tạp**.

Bài 7:

1. Chia nhỏ User Story thành các task

Task	Mô tả	Lý do lựa chọn	Ước lượng (T-shirt Size)
Task 1: Thiết kế giao diện form đăng ký	Tạo màn hình UI với các trường: tên, email, mật khẩu, xác nhận mật khẩu	Đây là bước đầu tiên để người dùng có thể nhập thông tin.	Small
Task 2: Kiểm tra dữ liệu nhập (Validation)	Xử lý logic kiểm tra dữ liệu: email hợp lệ, mật khẩu đủ mạnh, không để trống	Đảm bảo dữ liệu chính xác và an toàn trước khi gửi lên server.	Medium

Task 3: Viết API xử lý đăng ký	Tạo API backend để lưu thông tin người dùng vào cơ sở dữ liệu	Đây là phần quan trọng, liên quan đến logic nghiệp vụ và database.	Large
Task 4: Tích hợp bảo mật (mã hóa mật khẩu)	Mã hóa mật khẩu trước khi lưu, đảm bảo an toàn dữ liệu	Bảo mật là yếu tố bắt buộc trong tính năng đăng ký.	Medium
Task 5: Hiển thị thông báo kết quả đăng ký	Thông báo thành công hoặc lỗi cho người dùng	Giúp người dùng biết trạng thái thao tác, tăng trải nghiệm.	Small
Task 6: Kiểm thử chức năng đăng ký	Thực hiện test với nhiều trường hợp: dữ liệu hợp lệ, sai định dạng, trùng email	Đảm bảo tính năng hoạt động đúng và ổn định.	Medium

2. Lý do lựa chọn các task

- **Thiết kế giao diện:** cần thiết để người dùng có nơi nhập thông tin.
- **Validation:** tránh lỗi dữ liệu và nâng cao trải nghiệm.
- **API xử lý đăng ký:** cốt lõi để lưu thông tin người dùng.
- **Bảo mật:** bắt buộc trong ứng dụng có dữ liệu nhạy cảm.
- **Thông báo kết quả:** giúp người dùng biết thao tác thành công/thất bại.
- **Kiểm thử:** đảm bảo tính năng ổn định trước khi đưa vào sử dụng.

3. Ước lượng tổng thể

- Small: 2 task (UI, thông báo)
- Medium: 3 task (Validation, bảo mật, kiểm thử)
- Large: 1 task (API xử lý)

⇒ Tổng thể: **mức độ phức tạp vừa phải**, phù hợp để hoàn thành trong một Sprint.

Bài 8 :

Tính velocity theo sprint

- **Sprint 1:**

$$\text{Velocity}_1 = 5 + 8 = 13$$

(Chỉ tính các User Story hoàn thành; bỏ 3 điểm chưa hoàn thành)

- **Sprint 2:**

$$\text{Velocity}_2 = 13 + 5 = 18$$

(Bỏ 8 điểm chưa hoàn thành)

Velocity trung bình

$$\text{Velocity trung bình} = \frac{13 + 18}{2} = 15.5 \Rightarrow \text{nên lập kế hoạch 15–16 điểm}$$

Lập kế hoạch sprint tiếp theo bằng velocity

- **Ngưỡng cam kết:** 15–16 điểm Story.
- **Nguyên tắc chọn:** Ưu tiên các User Story phục vụ mục tiêu sprint, ít phụ thuộc mở; phối hợp Small/Medium để giảm rủi ro.

Ví dụ lựa chọn từ backlog

User Story	Điểm Story	Lý do chọn
US-01	8	Cốt lõi cho mục tiêu sprint, độ phức tạp vừa.
US-02	5	Bổ trợ trực tiếp cho US-01, rủi ro thấp.
US-03	3	Khép kín luồng chức năng, dễ kiểm thử.

Tổng điểm: $8 + 5 + 3 = 16$

Bài 9:

1. Ước lượng công việc bằng Story Points

- **User Story:** “Tạo công việc mới”
- **Ước lượng:** 5 điểm Story

- **Lý do:**

- Chức năng này yêu cầu thiết kế form nhập, kiểm tra dữ liệu, viết API lưu công việc và hiển thị thông báo.
- Mức độ phức tạp trung bình, không quá đơn giản nhưng cũng không quá lớn.

2. Thực hiện Planning Poker

Thành viên	Điểm đánh giá	Lý do
Dev A	5	Đánh giá trung bình, gồm UI + API cơ bản.
Dev B	3	Cho rằng chỉ cần form và validation đơn giản.
Dev C	8	Đánh giá cao vì có nhiều logic và kiểm thử.

- **Kết quả trung bình:**

$$(5 + 3 + 8)/3 = 5.3 \approx 5$$

- **Thảo luận sự khác biệt:**

- Dev B đánh giá thấp vì chỉ nhìn vào phần giao diện.
- Dev C đánh giá cao vì tính đến cả backend và kiểm thử.
- Nhóm thống nhất mức **5 điểm Story** để phản ánh độ phức tạp vừa phải.

3. Lập kế hoạch Sprint

- **Velocity của nhóm (Sprint trước):** 15 điểm Story.
- **Mục tiêu Sprint:** Hoàn thành tính năng “*Tạo công việc mới*” và “*Quản lý danh sách công việc*”.

User Stories được chọn:

1. **Tạo công việc mới** → 5 điểm
2. **Quản lý danh sách công việc** → 8 điểm
3. **Hiển thị chi tiết công việc** → 2 điểm

Tổng điểm: $5 + 8 + 2 = 15$

Bài 10:

1. Chia nhỏ User Story thành các task

Task	Mô tả	Lý do lựa chọn	Ước lượng (Story Points)
Task 1: Thiết kế giao diện giỏ hàng	Tạo form hiển thị sản phẩm trong giỏ, nút thêm/xóa, tổng tiền	Đây là bước đầu tiên để người dùng thao tác trực tiếp	3
Task 2: Xử lý logic thêm/xóa sản phẩm	Viết code xử lý khi người dùng thêm hoặc xóa sản phẩm	Đảm bảo giỏ hàng hoạt động đúng chức năng	5
Task 3: Viết API giỏ hàng	Tạo API backend để lưu trạng thái giỏ hàng vào database	Cần thiết để đồng bộ dữ liệu giữa client và server	8
Task 4: Tính toán tổng tiền và phí	Logic tính tổng giá trị đơn hàng, thuế, phí vận chuyển	Liên quan đến nhiều trường hợp nghiệp vụ	5
Task 5: Kiểm thử giỏ hàng	Test các trường hợp: thêm nhiều sản phẩm, xóa, cập nhật số lượng	Đảm bảo tính năng ổn định trước khi release	3

2. Lý do chia nhỏ như vậy

- **Tách UI và logic:** giúp phân công công việc rõ ràng giữa frontend và backend.
- **API riêng biệt:** đảm bảo dữ liệu giỏ hàng được lưu trữ và đồng bộ.
- **Tính toán tổng tiền:** là phần nghiệp vụ quan trọng, cần tách riêng để dễ kiểm thử.
- **Kiểm thử:** bắt buộc để đảm bảo chất lượng sản phẩm.

3. Lập kế hoạch Sprint

- **Tổng Story Points cho User Story “Giỏ hàng”:**

$$3 + 5 + 8 + 5 + 3 = 24$$

- **Velocity của nhóm (giả sử từ Sprint trước):** 20 điểm.
- **Kế hoạch Sprint:**

- Chọn các task quan trọng nhất để nằm trong Velocity:
 - Task 1 (3 SP) + Task 2 (5 SP) + Task 3 (8 SP) + Task 5 (3 SP) = **19 SP**
- Task 4 (5 SP) có thể để lại cho Sprint sau để không vượt quá Velocity.