

Bài 1:

- **User Story:** As a customer, I want create new task so that I can follow my work progress.
- **Ước lượng Story Point:** 5 điểm (Độ khó trung bình)
- **Lý do:** Với chức năng tạo công việc mới không quá khó và độ phức tạp mức trung bình, vì cần thiết kế form tạo, xử lý thao tác nhập liệu, validate thông tin, dạng chức năng cơ bản ít rủi ro.

Bài 2:

- **User Story:** As a customer, I want to pay for order so that I can pay for my products in my order cart.
- **Ước lượng Story Point:** 7 điểm
- **Lý do:** Cũng là 1 trong những chức năng cơ bản của 1 hệ thống bán hàng trực tuyến, nhưng thao tác cần phải liên hệ với người dùng, call đến giỏ hàng và những sản phẩm trong giỏ hàng hay thông tin sản phẩm cần thanh toán → khối lượng công việc & độ phức tạp tương đối, rủi ro trong giao dịch,

Bài 3:

- **User Story:** As a customer, I want add product to cart so that I can shopping in only one order
 - **Ước lượng Story Point:**
 - Mỗi thành viên chọn story point:
 - Nguyễn Văn A: 7 điểm - Có nhiều bước UI, validation và cập nhật dữ liệu
 - Nguyễn Văn B: 5 điểm – Tích hợp API, chủ yếu UI + xử lý client
 - Nguyễn Văn C: 10 điểm – Cần xử lý rủi ro khi số lượng lớn, logic tính giá, chiết khấu
 - Nguyễn Văn D: 8 điểm – Đồng thuận với Dev A, rủi ro vừa phải
 - Thảo luận:
 - Dev B thấy đơn giản → Story Point thấp
 - Dev C cần xử lý rủi ro, phức tạp → Story Point cao
 - Dev A thảo luận Dev B: chức năng nhiều bước, nhưng chung là mức độ Trung bình
- ⇒ Sau khi thảo luận, nhóm đồng thuận Story Point trung bình 8 điểm

Kết quả: Story Point vào khoảng 8 điểm

→ Tính năng bao gồm UI, xử lý nhập liệu + validate, rủi ro chưa quá phức tạp → mức độ khó trung bình cao, phù hợp 8 SP.

Bài 4:

- **User Story:** As a user, I want login into app, so that I can access into system.
- **Ước lượng Story Point:**
 - o **Mỗi thành viên chọn 1 story point**
 - Nguyễn Văn A: Xử lý UI, gọi API, bảo mật ↔ 8 điểm
 - Nguyễn Văn B: Chỉ UI + validate ↔ 5 điểm
 - Nguyễn Văn C: Cần xử lý OTP, tính bảo mật cao ↔ 13 điểm
 - Nguyễn Văn D: thảo luận với Dev A cùng đánh giá User Story đi đến đồng thuận ↔ 8 điểm
 - o **Thảo luận & Đồng thuận:**
 - Sau khi thảo luận nhóm thống nhất 8 điểm Story Point vì User Story có nhiều bước (UI, API, validate), yêu cầu bảo mật vừa phải (mật khẩu, OTP) và rủi ro trung bình. Sau thảo luận, 8 SP là mức trung gian hợp lý phản ánh tổng thể độ khó của User Story.

Bài 5:

1. Ước lượng mức độ phức tạp cho các User Story:

- Sử dụng Story Point:

User Story	Story Point (Ước lượng)
Đăng nhập người dùng	8
Đăng ký người dùng	5
Tạo mới công việc	5
Xem danh sách công việc	2
Xóa công việc	3

- Đánh giá và thảo luận Story Point bằng Planning Poker:
 - o Mỗi thành viên chọn thẻ Story Points riêng cho từng User Story.
 - o Mở thẻ cùng lúc → nếu có chênh lệch, thảo luận lý do (khối lượng, độ phức tạp, rủi ro).
 - o Sau thảo luận, thống nhất Story Points cuối cùng:
 - US01: 5 điểm Story Point
 - US02: 8 điểm Story Point
 - US03: 8 điểm Story Point
 - US04: 3 điểm Story Point
 - US05: 3 điểm Story Point
- Mục tiêu của Sprint này: Hoàn thành Nhóm chức năng xác thực là đăng nhập và đăng ký để người dùng có thể truy cập hệ thống

- Phân chia công việc và ước lượng thời gian:
 - Xử lý UI
 - Form đăng ký đăng nhập ~ 1 ngày
 - Giao diện báo lỗi/Thành công ~ 0,5 ngày
 - Xử lý nhập liệu + validate:
 - Kiểm tra dữ liệu nhập hợp lệ ~ 0,5 ngày
 - Validate xác thực (Mật khẩu, email) ~0,5 ngày
 - Backend/API:
 - API đăng ký, đăng nhập ~ 0,5 - 1 ngày
 - Bảo mật:
 - Mã hóa, xác thực OTP hạn chế đăng nhập sai ~ 1 ngày
 - Kiểm thử UI & chức năng ~ 1 ngày
- Sử dụng Valocity

Bài 6:

Dựa trên ứng dụng quản lý công việc

1. Xác định Mục tiêu Sprint:

“Hoàn thiện tính năng quản lý công việc cho người dùng”

2. Chọn ra User Story

- User Story 1: Đăng nhập ứng dụng
- User Story 2: Tạo mới công việc

➔ Hai chức năng được coi là tất yếu để người dùng có thể truy cập ứng dụng và quản lý công việc ↔ Đăng nhập thì mới truy cập được ứng dụng, Tạo được item công việc thì mới có đơn vị để người dùng theo dõi và quản lý

3. Phân chia User Story thành các task nhỏ

a) Đăng nhập ứng dụng:

- Tạo UI: Form đăng nhập, giao diện báo lỗi/thành công
- Xử lý dữ liệu nhập
- Xử lý xác thực đăng nhập, lưu database
- Thông báo đăng nhập

b) Tạo mới công việc:

- Xử lý UI:
 - Tạo model form chứa các trường thông tin cần thiết cho 1 công việc mới
 - Giao diện khi tạo mới thành công/lỗi
- Xử lý dữ liệu nhập hợp lệ
- Kiểm tra tồn tại
- Xử lý API và Database
- Thông báo thêm mới thành công/Lỗi

4. Ước lượng công việc cho mỗi User Story

- User Story 1: Small
 - ➔ Số lượng task được chia nhỏ vừa phải và chức năng không quá phức tạp
- User Story 2: Medium
 - ➔ Số lượng task được chia nhỏ tương tự US01 nhưng chức năng cần liên hệ đến cơ sở dữ liệu và kiểm tra công việc mới có hợp lệ không, đã tồn tại chưa

Bài 7:

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

- Chia nhỏ User Story:
 - o UI:
 - Xử lý giao diện form đăng ký chứa các trường thông tin cần thiết (**Medium**)
 - Tạo kèm giao diện khi đăng ký thành công/thất bại (**Small**)
 - ➔ Lý do: Để làm giao diện cho người dùng thao tác trước (form, thông báo).
 - o Xử lý nhập liệu:
 - Kiểm tra dữ liệu nhập hợp lệ (**Small**)
 - Validate xác thực (Mật khẩu, email) (**Medium**)
 - ➔ Lý do: để đảm bảo dữ liệu người dùng nhập đúng, tránh lỗi.
 - o Call API:
 - API đăng ký (**Medium**)
 - ➔ Lý do: Vì cần gửi dữ liệu lên server và nhận kết quả, là phần logic riêng.
 - o Bảo mật:
 - Mã hóa, xác thực OTP hạn chế đăng nhập sai (**Large**)
 - ➔ Lý do: Vì liên quan mật khẩu/OTP nên phải xử lý cẩn thận để an toàn.

Bài 9:

Dựa theo ứng dụng quản lý công việc

1. Ước lượng công việc bằng phương pháp Story Point:

- User Story: As a User, I want Create new Task so that I can follow and manage my tasks
- Ước lượng Story Point: 5 điểm
 - ➔ Lý do: Có **form nhập liệu** (title, description, deadline).
 - Cần **validate dữ liệu** (không bỏ trống, đúng format ngày).
 - Cần **gọi API tạo công việc**.
 - **Không quá phức tạp**, nhưng nhiều bước → mức **trung bình**.
 - 5 SP là phù hợp cho tính năng **CRUD** mức trung bình.

2. Áp dụng Planning Poker cho các User Story trong ứng dụng quản lý công việc (Phân tích cá nhân)

a) Chức năng đăng nhập ứng dụng

- **User Story:** As a user, I want login into app, so that I can access into system.

- **Ước lượng Story Point:**
 - o **Các thành viên đánh giá Story Point**
 - Dev 1: Xử lý UI, gọi API, bảo mật ↔ 8 điểm
 - Dev 2: Chỉ UI + validate ↔ 5 điểm
 - Dev 3: Cần xử lý OTP, tính bảo mật cao ↔ 13 điểm
 - Dev 4: thảo luận với Dev 1 cùng đánh giá User Story đi đến đồng thuận ↔ 8 điểm
 - o **Thảo luận & Đồng thuận:**
- Sau khi thảo luận nhóm thống nhất 8 điểm Story Point vì User Story có nhiều bước (UI, API, validate), yêu cầu bảo mật vừa phải (mật khẩu, OTP) và rủi ro trung bình. Sau thảo luận, 8 điểm Story Point là mức trung hợp lý phản ánh tổng thể độ khó của User Story
- b) Chức năng đăng ký người dùng
 - **User Story:** As a user, I want register an account so that I can use the app.
 - **Ước lượng Story Point:**
 - o Dev 1: UI + validate + call API ↔ 5 điểm
 - o Dev 2: Chỉ UI + validate ↔ 3 điểm
 - o Dev 3: Cần xử lý trùng email, validate nâng cao ↔ 8 điểm
 - o Dev 4: Thảo luận với Dev 1 ↔ đồng thuận ↔ 5 điểm
 - **Thảo luận & đồng thuận:**
Nhóm thống nhất **Story Point = 5** vì CRUD cơ bản, validate và gọi API, không quá phức tạp nhưng có nhiều bước → mức trung bình
- c) Tạo mới công việc
 - **User Story:** As a user, I want create new task so that I can follow/manage my tasks.
 - **Ước lượng Story Point:**
 - o Dev 1: Form + validate + API ↔ 5 điểm
 - o Dev 2: Chỉ form + validate ↔ 3 điểm
 - o Dev 3: Form + API + validate + deadline logic ↔ 8 điểm
 - o Dev 4: Thảo luận với Dev 1 → đồng thuận ↔ 5 điểm
 - **Thảo luận & đồng thuận:**
Nhóm thống nhất **Story Point = 5** vì nhiều bước, mức độ phức tạp trung bình → phù hợp cho tính năng CRUD cơ bản.
- d) Xem danh sách công việc
 - **Story:** As a user, I want view task list so that I can track my tasks.
 - **Ước lượng Story Point:**
 - o Dev A: Lấy dữ liệu + hiển thị → 2 điểm
 - o Dev B: Chỉ hiển thị → 1 SP
 - o Dev C: Hiển thị + lọc/sắp xếp → 3 SP
 - o Dev D: Thảo luận → đồng thuận → 2 SP
 - **Thảo luận & đồng thuận:**
Nhóm thống nhất **Story Point = 2** vì thao tác đơn giản, rủi ro thấp, không nhiều bước → mức thấp.

Bài 10:

1. Chọn User Story

- **User Story:** “As a User, I want to pay for my order so that I can complete the purchase in the app.”

2. Chia nhỏ User Story thành Task

Task	Lý do chia nhỏ	Story Point (SP)
Tạo giao diện thanh toán	Người dùng cần form nhập thông tin thanh toán	2 SP
Xử lý API thanh toán	Gửi dữ liệu thanh toán lên server, xác nhận giao dịch	3 SP
Kiểm tra giao dịch	Đảm bảo giao dịch thành công, xử lý lỗi nếu thất bại	2 SP
Hiển thị thông báo kết quả	Thông báo thành công/thất bại cho người dùng	1 SP
Tích hợp với ví thanh toán	Kết nối với các phương thức thanh toán, áp dụng mã giảm giá	2 SP

➔ **Tổng SP cho User Story:** 10 SP

➔ **Lý do chia nhỏ:**

- Giúp quản lý công việc dễ dàng, ước lượng chính xác hơn
- Mỗi task vừa sức, dễ phân công cho các thành viên
- Phát hiện sớm lỗi kỹ thuật hoặc UX

3. Lập kế hoạch Sprint

- **Velocity nhóm giả định:** 16 SP

- **Chọn User Stories:**

- “Thanh toán đơn hàng” → 10 SP
- “Tạo công việc mới” → 5 SP

- **Tổng SP:** $10 + 5 = 15 \text{ SP} \leq \text{Velocity (16 SP)}$ → khả thi

- **Tasks phân bổ:**

- Thanh toán đơn hàng: $2+3+2+1+2 = 10 \text{ SP}$
- Tạo công việc mới: chia task như “Tạo form”, “Validate dữ liệu”, “Call API” → 5 SP

➔ **Kết quả:**

- Sprint có thể hoàn thành 2 User Stories
- Tổng Story Points \leq Velocity → Sprint khả thi, cân bằng công việc cho nhóm