

## Session03

### Bài 1:

- User Story: "Là người dùng, tôi muốn tìm kiếm phim để hệ thống hiển thị danh sách phim phù hợp với từ khóa".
- Ước lượng Story Points: 5 điểm Story (mức độ khó trung bình)
- Lý do giải thích:
  - Tính năng tìm kiếm phim không quá phức tạp nhưng yêu cầu các bước quan trọng như:
    - Thiết kế ô tìm kiếm (search bar)
    - Xử lý sự kiện người dùng nhập từ khóa
    - Gửi yêu cầu đến API hoặc dữ liệu phim
    - Lọc kết quả dựa trên từ khóa
    - Hiển thị danh sách phim phù hợp
    - Xử lý trường hợp không có kết quả

### Bài 2:

- User Story: "Là người dùng, tôi muốn thanh toán đơn hàng một cách dễ dàng, tiện lợi".
- Ước lượng Story Points: 5 điểm Story (Mức độ khó trung bình).
- Lý do giải thích:
  - Tính năng thanh toán không quá phức tạp về giao diện, nhưng yêu cầu nhiều bước xử lý quan trọng như:
  - Thu thập thông tin đơn hàng và thông tin người dùng
  - Kiểm tra tính hợp lệ của thông tin thanh toán
  - Kết nối với cổng thanh toán (VD: Momo, VNPay hoặc thẻ ngân hàng)
  - Xử lý phản hồi từ cổng thanh toán (thành công / thất bại)
  - Hiển thị thông báo kết quả cho người dùng
  - Cập nhật trạng thái đơn hàng

### Bài 3:

1. User Store: "Là người dùng, tôi muốn thêm sản phẩm vào giỏ hàng để có thể mua hàng trong một đơn hàng duy nhất."
2. Điểm Story ước lượng

Thành viên	Lá bài chọn	điểm
Thành viên A	3	3 SP
Thành viên B	5	5 SP

Thành viên C	8	8 SP
--------------	---	------

3. Giải thích sự khác biệt:

- Thành viên A (3 SP): Nghĩ việc thêm sản phẩm vào giỏ hàng đơn giản, chỉ cần cập nhật UI và số lượng.
- Thành viên B (5 SP): Xem xét thêm logic kiểm tra tồn kho, tăng số lượng, thông báo → trung bình.
- Thành viên C (8 SP): Quan tâm đến rủi ro API, đồng bộ dữ liệu backend → phức tạp hơn.
- → Đồng thuận cuối cùng: 5 SP (mức trung bình), vì logic vừa phải, rủi ro thấp, UI đơn giản.

## Bài 4:

1. User Store: “Là một khách hàng, tôi muốn đăng nhập vào tài khoản ngân hàng trực tuyến của mình một cách an toàn để có thể quản lý tài chính”
2. Điểm Story ước lượng

Thành viên	Lá bài chọn	điểm
Thành viên A	3	3 SP
Thành viên B	8	8 SP
Thành viên C	8	8 SP

3. Giải thích sự khác biệt:

- Thành viên A (3 SP): Nghĩ đơn giản là làm giao diện nhập (User/Pass), chỉ cần kiểm tra đúng sai cơ bản.
- Thành viên B (8 SP): Xem xét thêm các logic bảo mật bắt buộc: mã hóa mật khẩu, tự động khóa sau 5 lần sai, timeout phiên làm việc.
- Thành viên C (13 SP): Quan tâm đến độ khó khi tích hợp xác thực 2 lớp (OTP) và sinh trắc học (Vân tay/FaceID) trên nhiều thiết bị.
- Đồng thuận cuối cùng: 8 SP (mức phức tạp cao), vì yêu cầu bảo mật của ngân hàng rất khắt khe, cần xử lý kỹ phần mã hóa và OTP, dù UI không quá nhiều.

## Bài 5:

1. Product Backlog & Story Pointsppp

User Story	Mô tả	Story Points
US1: Đăng nhập	Người dùng cần xác thực tài khoản để truy cập vào hệ thống quản lý.	5
US2: Tạo công việc	Người dùng tạo chi tiết 1 công việc mới	3
US3: Chỉnh sửa công việc	Người dùng chỉnh sửa nội dung công việc	5
US4: Xóa công việc	Người dùng xóa công việc khỏi danh sách	2
US5: Gắn nhãn ưu tiên	Người dùng gắn mức độ ưu tiên cho công việc	3
US6: Xem danh sách công việc	Hiển thị toàn bộ công việc theo trạng thái	8

## 2. Planning Poker – Ước lượng đồng thuận

Nhóm gồm 4 thành viên: A, B, C, D.

Ví dụ với User Story: Đăng nhập (US1)

- **A: 3** → nghĩ chỉ cần form và kiểm tra cơ bản.
- **B: 5** → xem xét thêm validation và xử lý lỗi.
- **C: 5** → đánh giá mức trung bình, có mã hóa mật khẩu.
- **D: 8** → lo ngại về các yêu cầu bảo mật nâng cao như OTP.

Kết quả sau thảo luận: thống nhất 5 Story Points.

Quy trình tương tự được áp dụng cho những User Story còn lại để tạo nên bảng ước lượng ban đầu.

## 3. Sprint Goal

“Hoàn thiện chức năng đăng nhập và các tính năng quản lý công việc cốt lõi: tạo, chỉnh sửa, xóa và xem danh sách công việc.”

## 4. Task Breakdown – Phân rã thành Tasks nhỏ

### US1 – Đăng nhập (5 points)

Task	Mô tả	Estimate
T1	Thiết kế giao diện form login	4 giờ
T2	Xử lý xác thực người dùng	5 giờ
T3	Mã hóa mật khẩu, kiểm tra dữ liệu đầu vào	3 giờ
T4	Hiển thị lỗi + xử lý trường hợp khóa tài khoản	3 giờ

### US2 – Tạo công việc (3 points)

Task	Mô tả	Estimate
T1	Thiết kế form thêm công việc	3 giờ
T2	kiểm tra dữ liệu nhập	2 giờ
T3	Lưu công việc mới vào data base	3 giờ

### US3 – Chỉnh sửa công việc (5 points)

Task	Estimate
Thiết kế màn sửa	4 giờ
Load dữ liệu cũ	3 giờ
Update vào DB	4 giờ

### US4 – Xóa công việc (2 points)

Task	Estimate
Nút xóa + confirm	2 giờ
Xóa DB	2 giờ

### US6 – Xem danh sách công việc (8 points)

Task	Estimate
Giao diện danh sách	6 giờ
Phân trang / lọc	8 giờ
Kết nối API	6 giờ

## 5. Sử dụng Velocity để lập kế hoạch Sprint

Velocity Sprint trước: **18 Story Points**

→ Sprint này chỉ nên chọn  $\leq 18$  SP.

**Sprint Backlog đề xuất:**

User Story	Story Points
US1 – Đăng nhập	5
US2 – Tạo công việc	3
US3 – Chỉnh sửa công việc	5
US4 – Xóa công việc	2
Tổng:	15 điểm

## Bài 6:

### 1. Xác định Mục tiêu Sprint

“Hoàn thành các tính năng cơ bản giúp người dùng tạo và quản lý công việc họ.”

#### Lý do lựa chọn mục tiêu:

- Đây là nhóm chức năng quan trọng nhất đối với ứng dụng quản lý công việc.
- Mang lại giá trị sử dụng ngay lập tức cho người dùng.
- Giúp thiết lập khung chức năng để mở rộng trong các Sprint kế tiếp.

### 2. Chọn User Stories từ Product Backlog

User Story	Mô tả	Lý do chọn
US1 – Tạo công việc mới	Người dùng có thể tạo một nhiệm vụ mới.	Đây là bước đầu tiên để bắt đầu quản lý công việc.
US2 – Xem danh sách công việc	“Là người dùng, tôi muốn xem danh sách công việc để quản lý tiến độ.”	Giúp người dùng theo dõi các công việc đã tạo → hoàn thiện luồng quản lý.

### 3. Phân chia User Stories thành tasks nhỏ

#### US1 – Tạo công việc mới

Task nhỏ	Mô tả
T1. Tạo giao diện form thêm công việc	Input: tên, mô tả, deadline
T2. Xử lý validate dữ liệu	Kiểm tra trống, định dạng ngày
T3. Lưu dữ liệu xuống database/API	Gửi request tạo task
T4. Hiển thị thông báo thành công/thất bại	Hiển thị alert/toast

## US2 – Xem danh sách công việc

Task nhỏ	Mô tả
T1. Thiết kế UI hiển thị danh sách	Giao diện danh sách công việc
T2. Kết nối API lấy danh sách	Gọi API lấy toàn bộ tasks
T3. Tạo chức năng lọc/sắp xếp (nếu có)	Lọc theo trạng thái, deadline
T4. Kiểm thử & xử lý lỗi	Kiểm tra hiển thị đúng danh sách

### 4. Ước lượng công việc (T-shirt Sizes)

User Story	Kích cỡ	Lý do
US1 – Tạo công việc	Medium (M)	Có form nhập + validate + API → mức độ trung bình, nhiều bước nhưng không phức tạp.
US2 – Xem danh sách công việc	Large (L)	Gồm UI + gọi API + hiển thị danh sách + lọc/sắp xếp → phức tạp hơn.

## Bài 7:

### 1. Chia nhỏ User Story thành các Task

Task	Mô tả công việc
T1. Thiết kế giao diện form đăng ký	Tạo giao diện gồm: tên, email, mật khẩu, xác nhận mật khẩu.
T2. Xử lý validate dữ liệu phía client	Kiểm tra email đúng định dạng, mật khẩu đủ mạnh, hai mật khẩu trùng nhau.
T3. Xử lý API đăng ký tài khoản (backend)	Nhận request, kiểm tra trùng email, mã hóa mật khẩu, lưu vào DB.
T4. Kết nối API với giao diện (frontend)	Gửi dữ liệu form lên server và nhận phản hồi.
T5. Hiển thị thông báo đăng ký thành công/thất bại	Thông báo lỗi khi email trùng, hoặc thông báo thành công.
T6. Điều hướng (redirect) sang trang đăng nhập	Chuyển người dùng sang màn hình login sau khi đăng ký thành công.

### 2. Lý do lựa chọn các Task

- Tách theo hành trình người dùng (User Workflow) → từ giao diện → validate → gửi dữ liệu → nhận kết quả → điều hướng.
- Mỗi task nhỏ, rõ ràng, có thể hoàn thành độc lập và không quá phức tạp.

- Đảm bảo logic đầy đủ để hoàn thành User Story, vì User Story “đăng ký tài khoản” cần: Form giao diện, Validate, Lưu tài khoản, Thông báo, Điều hướng

=> Các task trên bao phủ toàn bộ yêu cầu.

### 3. Ước lượng độ phức tạp (T-shirt Sizes)

Task	Ước lượng (T-shirt Size)	Lý do
T1. Thiết kế form đăng ký	S	UI đơn giản, dễ làm.
T2. Validate phía client	M	Nhiều quy tắc: email, mật khẩu, confirm password.
T3. API đăng ký (backend)	L	Liên quan DB, mã hóa mật khẩu, kiểm tra trùng email.
T4. Kết nối API	M	Gửi request + xử lý phản hồi.
T5. Hiển thị thông báo	S	Chỉ cần alert/toast đơn giản.
T6. Điều hướng sang login	XS	Rất đơn giản, chỉ redirect.

## Bài 8:

### 1. Dữ liệu Sprint đã hoàn thành

#### Sprint 1

User Story	Story Points	Hoàn thành
US1	5	có
US2	8	có
US3	3	không

- Velocity Sprint 1: Tổng điểm các User Story hoàn thành =  $5 + 8 = 13$  điểm

#### Sprint 2

User Story	Story Points	Hoàn thành
US4	13	có

User Story	Story Points	Hoàn thành
US5	5	có
US6	8	không

- Velocity Sprint 2: Tổng điểm các User Story hoàn thành =  $13 + 5 = 18$  điểm

## 2. Tính Velocity trung bình

-Nhóm có thể hoàn thành khoảng **16 điểm Story** trong Sprint tiếp theo.

## 3. Lập kế hoạch Sprint tiếp theo (Sprint 3)

Giả sử Product Backlog Sprint 3 có các User Stories:

User Story	Story Points
US7	5
US8	8
US9	3
US10	13
US11	2

**Phương án 1:** Tổng điểm = 16

Chọn: US8 (8) + US7 (5) + US9 (3) = 16

STT	User Story	Story Points	Trạng thái dự kiến
1	US8	8	Chưa bắt đầu
2	US7	5	Chưa bắt đầu
3	US9	3	Chưa bắt đầu

**Phương án 2:** Tổng điểm = 15

- Chọn: US10 (13) + US11 (2) = 15

STT	User Story	Story Points	Trạng thái dự kiến
1	US10	13	Chưa bắt đầu



STT	User Story	Story Points	Trạng thái dự kiến
2	US11	2	Chưa bắt đầu

#### 4. Nhận xét

1. Velocity Sprint 1 = 13, Sprint 2 = 18 → Velocity trung bình  $\approx 16$  điểm
2. Nhóm có thể hoàn thành **15–16 điểm Story** trong Sprint tiếp theo.
3. Tùy theo ưu tiên, nhóm có thể:
  - Hoàn thành nhiều User Stories nhỏ (Phương án 1)
  - Hoàn thành ít User Stories lớn (Phương án 2)

## Bài 9:

### 1. Mở đầu

Là Product Owner, tôi đang chuẩn bị Sprint Planning để triển khai một số tính năng mới cho ứng dụng quản lý công việc. Các tính năng chính cần triển khai trong Sprint tiếp theo bao gồm:

- **User Story 1:** Tạo công việc mới
- **User Story 2:** Quản lý danh sách công việc
- **User Story 3:** Sửa/Xóa công việc
- **User Story 4:** Tìm kiếm và lọc công việc
- **User Story 5:** Thêm thông báo nhắc nhở công việc

Mục tiêu là ước lượng công việc, xác định khả năng hoàn thành dựa trên **Velocity** và lập kế hoạch Sprint hợp lý.

### 2. Ước lượng công việc bằng Story Points

**User Story ví dụ: Tạo công việc mới**

“As a user, I want to create a new task so that I can manage my work efficiently.”

**Ước lượng Story Points:** 5 điểm

**Lý do chọn 5 điểm:**

- Tính phức tạp trung bình: Form tạo công việc có các trường cơ bản (Tên, Mô tả, Ngày hết hạn, Trạng thái).
- Không có tích hợp phức tạp như gửi email hoặc thông báo push.
- Cần kiểm tra validation dữ liệu và UI/UX.

Do đó, Story này không quá đơn giản (2–3 điểm) nhưng cũng không phức tạp quá (8–13 điểm), nên chọn 5 điểm.

### 3. Planning Poker

#### Các User Stories và ước lượng

Giả sử nhóm gồm 3 thành viên, chúng tôi áp dụng Planning Poker để ước lượng từng Story:

User Story	Thành viên A	Thành viên B	Thành viên C	Kết quả Story Points
Tạo công việc mới	5	5	5	5
Quản lý danh sách công việc	8	5	8	8
Sửa/Xóa công việc	3	5	3	3
Tìm kiếm và lọc công việc	5	8	5	5
Thêm thông báo nhắc nhở công việc	8	8	8	8

#### Thảo luận:

- Thành viên B cho rằng quản lý danh sách công việc ít phức tạp hơn, nhưng sau khi thảo luận, cả nhóm thống nhất **8 điểm** vì cần tính năng phân trang, sắp xếp và filter.
- Các Story khác nhận Story Points đồng thuận nhanh chóng.

### 4. Velocity của nhóm

Giả sử nhóm đã có dữ liệu Velocity từ các Sprint trước:

Sprint	User Stories hoàn thành	Velocity
Sprint 1	13 điểm	13
Sprint 2	18 điểm	18

**Velocity trung bình =  $(13 + 18) / 2 = 15.5 \approx 16$  điểm**

-Sprint tiếp theo nhóm có thể hoàn thành khoảng 16 điểm Story.

## 5. Lập kế hoạch Sprint tiếp theo

Dựa trên Velocity trung bình **16 điểm**, nhóm lựa chọn các User Stories từ Product Backlog:

User Story	Story Points
Tạo công việc mới	5
Sửa/Xóa công việc	3
Tìm kiếm và lọc công việc	5
Tổng cộng	13

- Tổng điểm  $13 \leq$  Velocity trung bình 16  $\rightarrow$  Sprint vừa sức.
- Còn lại các Story phức tạp hơn như Quản lý danh sách công việc (8 điểm) và Thông báo nhắc nhở (8 điểm) sẽ được lên kế hoạch cho Sprint sau.

### Kế hoạch Sprint 3:

STT	User Story	Story Points	Trạng thái dự kiến
1	Tạo công việc mới	5	Chưa bắt đầu
2	Sửa/Xóa công việc	3	Chưa bắt đầu
3	Tìm kiếm và lọc công việc	5	Chưa bắt đầu

## 6. Kết luận

1. Sử dụng **Story Points** giúp đánh giá mức độ phức tạp từng User Story.
2. **Planning Poker** đảm bảo nhóm đồng thuận và cân nhắc các quan điểm khác nhau.
3. **Velocity** giúp xác định khả năng hoàn thành công việc trong Sprint tiếp theo.
4. Sprint tiếp theo được lập kế hoạch hợp lý, tránh quá tải, đảm bảo nhóm có thể hoàn thành các User Stories đã chọn.

## Bài 10:

### 1. Mở đầu

- Trong ứng dụng bán hàng trực tuyến, nhóm Scrum cần triển khai **User Story** sau trong Sprint tiếp theo:

**User Story:** “As a user, I want to have a shopping cart so that I can add products and make purchases easily.”

Mục tiêu là: chia nhỏ User Story thành các task cụ thể, ước lượng độ phức tạp từng task bằng Story Points, và lập kế hoạch Sprint phù hợp với Velocity của nhóm.

## 2. Chia nhỏ User Story thành Task

Để dễ quản lý và ước lượng, User Story “Tạo giỏ hàng” được chia thành các task sau:

STT	Task	Mô tả	Story Points	Lý do chia task
1	Thiết kế UI giỏ hàng	Tạo giao diện hiển thị giỏ hàng, danh sách sản phẩm, số lượng, tổng tiền	3	Task UI tách biệt giúp designer và dev frontend làm song song
2	Xây dựng API giỏ hàng	Backend xử lý thêm/xóa sản phẩm, tính tổng tiền	5	Task backend phức tạp, cần đảm bảo dữ liệu chính xác
3	Tích hợp frontend với API giỏ hàng	Kết nối UI với API, hiển thị sản phẩm thêm/xóa, cập nhật tổng tiền	3	Task tích hợp giúp kiểm thử từng phần hoạt động liên kết
4	Kiểm thử giỏ hàng	Viết test case, kiểm tra UI, API và tích hợp	2	Task QA riêng biệt giúp đảm bảo chất lượng sản phẩm
5	Xử lý thông báo và UX	Hiển thị thông báo khi thêm/xóa sản phẩm, xử lý trải nghiệm người dùng	2	Task UX riêng để tăng trải nghiệm mà không ảnh hưởng API hay UI chính

Tổng Story Points cho User Story:  $3 + 5 + 3 + 2 + 2 = 15$  điểm

- Giúp nhóm dễ dàng ước lượng và phân công công việc.
- Mỗi task có trách nhiệm riêng, dễ theo dõi tiến độ.
- Giảm rủi ro vì có thể triển khai và test từng phần riêng biệt.

## 3. Lập kế hoạch Sprint dựa trên Velocity

Giả sử nhóm có dữ liệu **Velocity trung bình** từ các Sprint trước:

Sprint	Velocity
Sprint 1	13
Sprint 2	18

-Velocity trung bình =  $(13 + 18) / 2 = 15.5 \approx 16$  điểm

Nhóm có thể lập kế hoạch Sprint tiếp theo tổng số Story Points  $\approx 16$  điểm.

### Lựa chọn User Story và task cho Sprint

- Chọn **User Story Giỏ hàng** (15 điểm) → phù hợp với Velocity trung bình.
- Sprint này tập trung hoàn thành toàn bộ User Story để đảm bảo tính năng có thể demo.

STT	Task	Story Points	Trạng thái dự kiến
1	Thiết kế UI giỏ hàng	3	Chưa bắt đầu
2	Xây dựng API giỏ hàng	5	Chưa bắt đầu
3	Tích hợp frontend với API giỏ hàng	3	Chưa bắt đầu
4	Kiểm thử giỏ hàng	2	Chưa bắt đầu
5	Xử lý thông báo và UX	2	Chưa bắt đầu

Tổng điểm =  $15 \leq 16$  (Velocity) → Sprint khả thi, không quá tải.