

Bài 1:

## 1. Bối cảnh

- **Scrum Master:** Bạn (Sinh viên A)
- **Nhóm Dev:**
  - **Dev1 – Minh:** Phụ trách giao diện màn hình đăng nhập.
  - **Dev2 – Lan:** Xử lý API đăng ký người dùng.
  - **Dev3 – Huy:** Tích hợp hệ thống thông báo (push notification).
- **Sản phẩm:** Ứng dụng di động “HealthyLife”.
- **Sprint:** Sprint 3 – mục tiêu hoàn thiện chức năng Onboarding + Authentication.

## 2. Mục tiêu cuộc họp Daily Scrum

- Cập nhật tiến độ trong 24 giờ qua.
- Kế hoạch 24 giờ tới.
- Xác định blockers (trở ngại) và yêu cầu hỗ trợ.
- Đảm bảo cuộc họp không quá **15 phút** và chỉ tập trung vào công việc của Sprint.

## 3. Kịch bản cuộc họp Daily Scrum (mẫu hoàn chỉnh)

### Scrum Master bắt đầu cuộc họp

"Chào mọi người, chúng ta bắt đầu Daily Scrum trong 15 phút.

Mỗi bạn lần lượt trả lời 3 câu hỏi:

**1: Hôm qua đã làm gì?**

**2: Hôm nay sẽ làm gì?**

**3: Có trở ngại gì không?**

Mời Minh báo cáo trước."

### Dev1 — Minh

#### Hôm qua:

- Hoàn thành bố cục UI màn hình Login và phần xử lý validation email/password.
- Gửi Pull Request cho team review.

#### Hôm nay:

- Sửa lại UI theo comment review.
- Bắt đầu làm animation chuyển cảnh từ màn hình Onboarding sang Login.

#### Trở ngại:

- Chưa thống nhất tiêu chuẩn màu sắc và font chữ trong Design System.
- Cần Designer gửi bản final.

### **Scrum Master phản hồi:**

"Ok Minh, mình sẽ ping Designer sau họp và yêu cầu gửi Design System hoàn chỉnh trong hôm nay."

### **Dev2 — Lan**

#### **Hôm qua:**

- Tạo API đăng ký tài khoản (Signup).
- Tích hợp Firebase Authentication ở backend.

#### **Hôm nay:**

- Viết test cho API Signup.
- Tích hợp API với UI của Minh.

#### **Trở ngại:**

- Chưa có mock UI chính xác vì Minh còn đang chỉnh.

### **Scrum Master:**

"Lan cứ hoàn thành phần test trước. Khi Minh update UI cuối buổi sáng, cả hai sync lại."

### **Dev3 — Huy**

#### **Hôm qua:**

- Cấu hình Firebase Cloud Messaging.
- Hoàn thiện chức năng gửi thông báo thử nghiệm.

#### **Hôm nay:**

- Tạo màn hình cài đặt thông báo trong app (bật/tắt).
- Viết tài liệu hướng dẫn cách generate token.

#### **Trở ngại:**

- Bị lỗi khi test thông báo trên iOS do thiếu certificate APNs.

### **Scrum Master:**

"Huy gửi lỗi cho mình. Sau họp, mình sẽ làm việc với DevOps để tạo certificate mới."

## **4. Scrum Master tổng kết cuộc họp**

**"Cảm ơn mọi người."**

**Các blocker hôm nay gồm:**

- Minh thiếu Design System → Scrum Master liên hệ Designer.
- Lan cần UI ổn định → chờ Minh chỉnh sửa sáng nay.
- Huy thiếu certificate iOS → Scrum Master làm việc với DevOps.

### Nhắc lại mục tiêu Sprint:

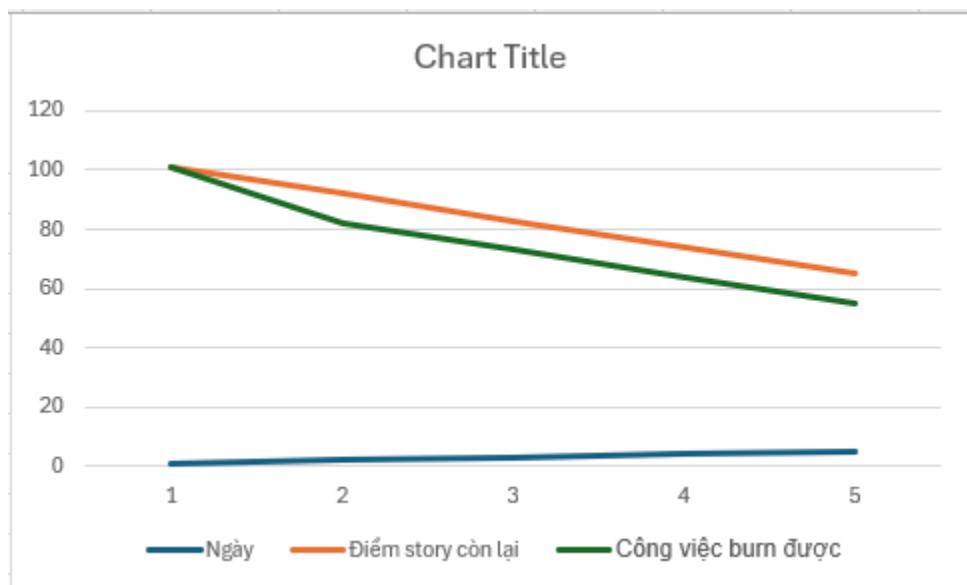
- Hoàn thiện Onboarding + Authentication.  
Mọi người nhớ cập nhật task trên Jira/Backlog trước 11h nhé.  
Cuộc họp đến đây kết thúc."

## 5. Kết quả đạt được sau cuộc họp

- Nhóm nắm rõ tiến độ chung.
- Blockers được phát hiện sớm và có người chịu trách nhiệm xử lý.
- Giữ đúng thời gian, đúng quy trình Daily Scrum.

Bài 2:

### 1. Sơ đồ burndown chart



### 2. Mô phỏng tiến độ 5 ngày đầu

Ngày	Điểm story còn lại	Công việc burn được
1	100	0
2	90	-10
3	80	-10
4	70	-10
5	60	-10

### **3. Giải thích và đánh giá tiến độ trên Burndown Chart**

#### **3.1. Nhóm có đang đi đúng hướng không?**

→ **Có.**

Nhóm đang burn đều 10 điểm mỗi ngày.

Sprint 14 ngày → trung bình cần burn ~7 điểm/ngày.

Nhóm đang burn **nhiều hơn mức cần thiết** → **vượt tiến độ**.

#### **3.2. Điều này cho thấy gì?**

- Nhóm phối hợp tốt.
- Không có trở ngại lớn.
- Các nhiệm vụ được chia hợp lý.
- Burndown Chart cho thấy xu hướng **giảm tuyến tính**, rất lý tưởng trong Scrum.

#### **3.3. Scrum Master cần làm gì?**

- Tiếp tục theo dõi để đảm bảo không có sự “bung cháy sớm” (burn nhanh rồi chững lại).
- Đảm bảo khối lượng công việc được phân phối đều.
- Kiểm tra xem liệu nhóm có đang ước lượng đúng hay đang làm các task dễ trước.

### **4. Phân tích tình huống Burndown Chart KHÔNG giảm đều**

**Tình huống giả định:**

- Đến ngày 5, lượng story còn lại không phải 60 mà là **40** (tức burn được 60 điểm trong 5 ngày).

#### **4.1. Dấu hiệu nhận biết**

Burndown Chart sẽ có đoạn giảm mạnh đột ngột → **burn quá nhanh**.

#### **4.2. Nguy cơ**

- Nhóm làm các task dễ trước → các task khó dồn về cuối Sprint.
- Có thể ước lượng sai (underestimate).
- Có thể nhóm đang “đóng task ảo” (chưa hoàn thành Definition of Done).
- Có thể backlog chưa được refine tốt.

#### **4.3. Scrum Master cần làm gì?**

##### **1. Kiểm tra Definition of Done (DoD)**

- Đảm bảo các task thực sự hoàn thành (code + review + test).
- Tránh việc chỉ “đẩy lên Done cho kịp”.

##### **2. Tổ chức một phiên họp chỉnh sửa lại Sprint Backlog**

- Xem lại độ khó các task còn lại.
- Xếp lại mức ưu tiên và effort.

### 3. Làm rõ lý do tại sao burn nhanh

Hỏi đội nhóm trong Daily Scrum:

- “Có task nào chưa hoàn thành đúng chuẩn?”
- “Có task nào còn phụ thuộc chưa xử lý?”

### 4. Điều chỉnh ngay nếu phát hiện rủi ro

- Chia nhỏ task còn lại dễ test hơn.
- Giảm rủi ro dồn việc về cuối Sprint.

Bài 3:

## Tổ chức một cuộc họp Daily Scrum hiệu quả

### Bước 1: Mô phỏng một cuộc họp Daily Scrum cho một ngày trong Sprint

Bối cảnh:

- Sprint 2 tuần (14 ngày).
- Nhóm đang xây dựng ứng dụng “TaskFlow”.
- Cuộc họp diễn ra vào ngày thứ 6 trong Sprint.
- Thành viên tham gia:
  - **Scrum Master:** Bạn
  - **Dev1 – Minh:** Frontend
  - **Dev2 – Lan:** Backend
  - **Dev3 – Huy:** Mobile

#### Cuộc họp Daily Scrum (15 phút):

##### Dev1 – Minh

Hôm qua:

- Hoàn thiện UI màn hình danh sách công việc và xử lý phần animation khi cuộn.

Hôm nay:

- Tích hợp API “Get Task List” từ backend.
- Tối ưu hiệu suất render.

Cản trở:

- API trả về cấu trúc JSON khác với tài liệu ban đầu nên chưa map dữ liệu chuẩn.

## Dev2 – Lan

### Hôm qua:

- Xây dựng API tạo công việc mới (CreateTask).
- Viết test cho API.

### Hôm nay:

- Sửa lại cấu trúc JSON trả về cho phù hợp với UI.
- Hỗ trợ Minh debug phần mapping dữ liệu.

### Cản trở:

- Chưa rõ một số thuộc tính cần thiết, vì tài liệu yêu cầu từ khách hàng chưa đầy đủ.

## Dev3 – Huy

### Hôm qua:

- Làm chức năng push notification khi tạo task mới.

### Hôm nay:

- Tối ưu token lưu trữ trong app.
- Tích hợp thông báo với backend.

### Cản trở:

- Hệ thống notification bị delay trên iOS, cần DevOps kiểm tra server.

## Bước 2: Mô phỏng cách nhóm giải quyết vấn đề trong cuộc họp

### Vấn đề:

- API trả về dữ liệu sai cấu trúc → frontend không tích hợp được.
- Tài liệu yêu cầu từ khách hàng thiếu rõ ràng (thiếu thuộc tính cần thiết).
- Notification trên iOS bị delay.

### Thảo luận trong Daily Scrum:

#### Scrum Master:

"Anh chị ghi lại blockers. Lan sau cuộc họp hãy sync nhanh với Minh để thống nhất cấu trúc JSON chuẩn và update lên cùng trong hôm nay."

Còn phần thiếu tài liệu từ khách, tôi sẽ liên hệ Product Owner để bổ sung ngay trong buổi sáng.

Huy gửi log lỗi đêm qua cho tôi, tôi chuyển cho DevOps kiểm tra server để fix việc delay thông báo iOS."

### Kết quả:

- Backend + frontend hẹn nhau 30 phút sau họp để chỉnh API.
- Scrum Master làm việc trực tiếp với PO để hoàn thiện tài liệu.
- DevOps nhận được yêu cầu hỗ trợ để xử lý lỗi thông báo chậm.

### **Bước 3: Lời khuyên để giữ cuộc họp ngắn gọn và hiệu quả**

Để tránh cuộc họp kéo dài quá 15 phút, Scrum Master nên áp dụng:

#### **1. Chỉ trả lời 3 câu hỏi chuẩn**

- Hôm qua làm gì?
- Hôm nay làm gì?
- Có vướng mắc gì không?

→ Không được kể lan man, không demo trong họp.

#### **2. Không giải quyết vấn đề trong cuộc họp**

Daily Scrum chỉ **nêu** vấn đề, không giải thích hoặc debug tại chỗ.

Giải quyết sau khi họp (tạo meeting nhỏ nếu cần).

#### **3. Đứng khi họp**

“Daily Standup” – đứng họp giúp cuộc họp tự động ngắn hơn.

#### **4. Giới hạn thời gian cho mỗi người (1–2 phút)**

Scrum Master có thể nhắc:

“Còn 5 phút nữa là hết họp, mọi người giữ câu trả lời ngắn gọn giúp nhé.”

#### **5. Sử dụng board (Jira / Trello)**

Nhóm nhìn vào board → tập trung vào tiến độ Sprint, không lạc chủ đề.

### **Bước 4: Vai trò của Scrum Master trong cuộc họp**

Trong Daily Scrum, Scrum Master không báo cáo tiến độ, mà có nhiệm vụ:

#### **1. Giữ cuộc họp đúng quy trình**

- Nhắc mọi người trả lời 3 câu hỏi.
- Giữ cuộc họp trong 15 phút.
- Không để mọi người thảo luận quá sâu vào kỹ thuật.

#### **2. Loại bỏ chướng ngại (Impediments)**

Sau cuộc họp, Scrum Master:

- Liên hệ Product Owner để bổ sung tài liệu.

- Điều phối giữa frontend – backend để thống nhất API.
- Làm việc với DevOps khi có lỗi hệ thống.
- Tạo buổi họp nhỏ để giải quyết blockers.

### 3. Đảm bảo tính minh bạch

- Cập nhật Sprint Board.
- Nhắc mọi người cập nhật trạng thái task.
- Đảm bảo mọi người hiểu Sprint Goal.

### 4. Khuyến khích nhóm tập trung vào mục tiêu Sprint

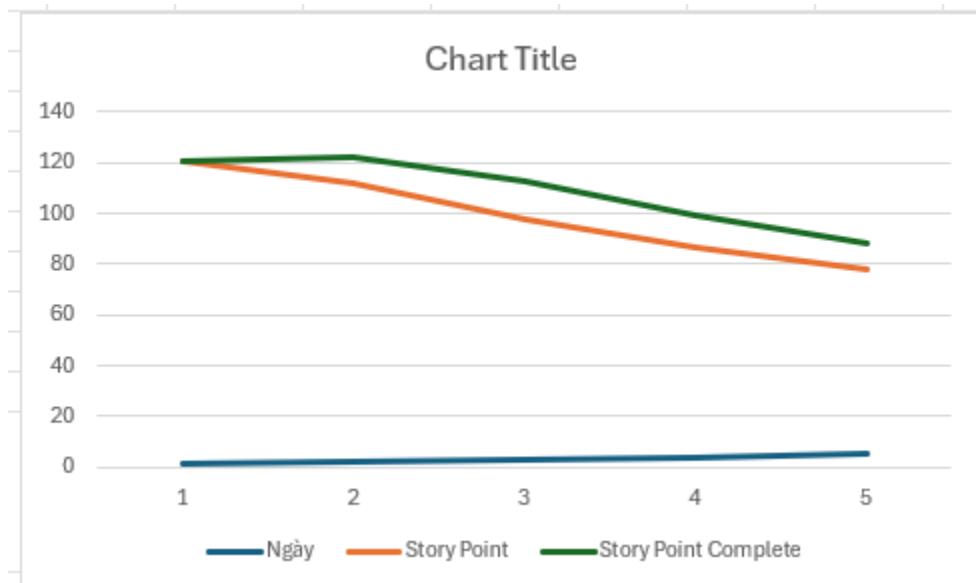
Scrum Master nhắc lại:

“Mục tiêu Sprint của chúng ta là hoàn thiện chức năng quản lý công việc. Mọi thứ khác để sau Sprint.”

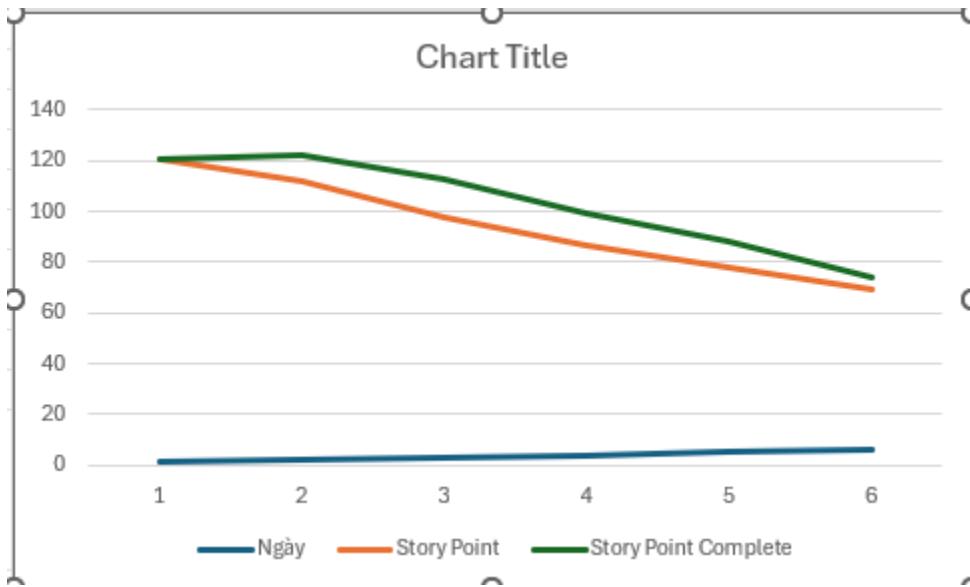
Bài 4:

**Bảng dữ liệu ban đầu:**

Ngày	Story Point	Story Point Complete
1	120	0
2	?	?
3	?	?
4	?	?
5	?	?
6	?	?
7	?	?
8	?	?
9	?	?
10	?	?



Day 6 có trục trặc



#### Đánh giá Burndown Chart:

- Đường Actual đột ngột **ngang** (gần như không giảm).
- Khoảng cách giữa Ideal Line và Actual Line mở rộng.
- Sprint bắt đầu **có nguy cơ không đạt mục tiêu**.

#### Cách nhóm điều chỉnh để khắc phục:

##### 1. Scrum Master tổ chức meeting nhanh (ngay sau Daily Scrum)

- Xác định nguyên nhân: thiếu người + API backend chậm.

##### 2. Điều phối tài nguyên

- Điều chuyển 1 dev từ module khác sang hỗ trợ backend.

##### 3. Re-prioritize (điều chỉnh ưu tiên công việc)

- Tập trung trước vào các task có thể làm ngay.
- Đẩy bớt các task không quan trọng xuống cuối Sprint.

##### 4. Cập nhật lại Sprint Backlog

- Ước lượng lại effort cho các task tồn đọng.
- Sắp xếp lại để đảm bảo vẫn đạt Sprint Goal cốt lõi.

#### Kết quả của điều chỉnh trên biểu đồ:

- Từ ngày 7 trở đi, Actual Line sẽ giảm mạnh trở lại nếu nhóm xử lý kịp.

Scrum Master dùng Burndown Chart để:

##### 1. Theo dõi tiến độ hàng ngày

- So sánh Actual Line với Ideal Line.
- Nhận biết tiến độ nhanh/chậm.
- Kiểm tra task nào đang bị chậm hoặc bị “kẹt”.

## 2. Phát hiện vấn đề sớm

Khi thấy:

- Đường không giảm
- Đường đi ngang
- Đường tăng (công việc bị thêm)
- Khoảng cách giữa hai line mở rộng

→ Đây là dấu hiệu Sprint đang gặp nguy hiểm.

## 3. Hành động nhanh để gỡ chướng ngại

Scrum Master cần:

- Làm việc với Product Owner để clarify yêu cầu.
- Hỗ trợ nhóm loại bỏ blockers.
- Điều phối tài nguyên nếu nhóm thiếu người.
- Điều chỉnh Sprint Backlog khi cần.

## 4. Giữ nhóm tập trung vào Sprint Goal

Scrum Master nhắc nhở:

“Không làm những việc ngoài Sprint Goal.

Tập trung vào những task đầy tiến độ về đúng quỹ đạo.”

Bài 5:

Xây dựng kanban board

Kanban Board		
To Do	In Progress	Done
Thiết kế giao diện ứng dụng		
Phát triển API đăng nhập		
Tạo cơ sở dữ liệu người dùng		
Kiểm tra bảo mật API		
Tích hợp tính năng thanh toán		
Viết tài liệu hướng dẫn sử dụng		
Kiểm thử giao diện người dùng (UI Testing)		

## 2. Đặt giới hạn WIP (Work In Progress)

Nhóm quyết định:

- **Giới hạn WIP cho In Progress = 3 task**
- Các cột khác không cần giới hạn.

### Tại sao giới hạn WIP quan trọng?

- Ngăn tình trạng mỗi thành viên làm quá nhiều việc nhưng không hoàn thành được việc nào.
- Tập trung hoàn thành công việc trước khi bắt đầu việc mới → tăng năng suất.
- Giảm "context switching" (nhảy việc), cải thiện chất lượng code.
- Giúp Scrum Master nhanh chóng nhận diện tắc nghẽn (bottleneck).
- Khuyến khích sự cộng tác: khi In Progress đầy, các thành viên buộc phải hỗ trợ nhau hoàn thiện task hiện có.

## 3. Mô phỏng tình huống bị quá tải và tối ưu hóa Kanban Board

### Tình huống

Trong tuần thứ 2 của Sprint, nhóm rơi vào tình trạng:

#### Cột In Progress (đang bị quá tải)

- Thiết kế giao diện
  - Phát triển API đăng nhập
  - Kiểm tra bảo mật API
  - Tích hợp thanh toán
  - Viết tài liệu hướng dẫn
- => Tổng: **5 nhiệm vụ đang làm** (vượt giới hạn 3)

### Hậu quả

- Tiến độ chậm
- Thành viên không ai hoàn thành task
- Nhiều lỗi xuất hiện nhưng chưa được xử lý
- Không có nhiệm vụ nào chuyển sang Done

### Cách Scrum Master điều chỉnh và tối ưu Kanban Board

#### 1. Nhắc nhóm tuân thủ WIP limit

- Scrum Master yêu cầu dừng việc kéo task mới vào In Progress.

- Bắt buộc cả nhóm tập trung xử lý 3 task đang gần hoàn thành nhất.

## 2. Tổ chức mini-meeting (5 phút) để phân tích tắc nghẽn

Ví dụ:

- Task API đăng nhập bị chậm do thiếu dữ liệu từ backend → cần hỗ trợ ngay.
- Task thiết kế giao diện gần xong → ưu tiên hoàn thiện.

## 3. Thay đổi quy trình làm việc

- **Ưu tiên hoàn thành thay vì bắt đầu mới** (“Stop Starting – Start Finishing”)
- Cho phép 2 người cùng xử lý 1 task quan trọng (pair programming)
- Thêm cột **Review/Testing** nếu cần (Kanban mở rộng)

## 4. Cập nhật Kanban Board sau tối ưu

### Cột In Progress (giảm còn đúng 3 task)

- Thiết kế giao diện (gần xong)
- API đăng nhập (gặp vấn đề cần phối hợp)
- Kiểm tra bảo mật API

Các task còn lại chuyển lại về To Do hoặc Review tùy mức độ ưu tiên.

## 4. Vai trò của Scrum Master khi sử dụng Kanban Board (Bước 4)

Scrum Master có trách nhiệm:

### 1. Hướng dẫn nhóm sử dụng Kanban đúng cách

- Giải thích ý nghĩa từng cột
- Giải thích WIP limit
- Nhắc nhở cập nhật board hằng ngày

### 2. Theo dõi và phát hiện bottleneck

- Thấy cột In Progress bị đầy → cảnh báo ngay
- Nếu task đứng yên quá lâu → điều tra nguyên nhân

### 3. Tối ưu quy trình

- Đề xuất thêm cột (Review/Testing/Blocked) nếu cần
- Yêu cầu đổi cách làm khi nhóm không hiệu quả
- Hỗ trợ thành viên giải quyết các trở ngại (impediments)

## 4. Tạo môi trường cộng tác

- Khuyến khích kết hợp pair programming
- Giảm tình trạng ai làm nấy
- Đảm bảo mọi người hiểu nhiệm vụ chung của Sprint

## 5. Đảm bảo Kanban Board được cập nhật hằng ngày

Scrum Master không cập nhật thay nhóm, mà đảm bảo **nhóm tự cập nhật** mỗi khi:

- Bắt đầu task
- Gặp vấn đề
- Hoàn thành công việc

Bài 6:

### 1. Kế hoạch Daily Scrum

#### 1.1 Cách tổ chức một buổi Daily Scrum chuẩn

- Cuộc họp diễn ra **mỗi ngày**, đúng **thời điểm cố định** (ví dụ 9:00 sáng).
- Thời lượng tối đa **15 phút** và phải diễn ra tại cùng một địa điểm (hoặc một link nếu họp online).
- Tất cả thành viên nhóm phát triển phải tham gia: Developer, QA, UI/UX, v.v.
- Scrum Master điều phối cuộc họp nhưng **không phải là người trình bày**.
- Mỗi thành viên trả lời 3 câu hỏi theo thứ tự, không thảo luận lan man.
- Nếu có vấn đề phát sinh → đưa vào After Meeting để xử lý sau.

#### 1.2 Ba câu hỏi chính trong Daily Scrum

1. **Hôm qua tôi đã làm gì để giúp nhóm đạt mục tiêu Sprint?**
2. **Hôm nay tôi sẽ làm gì để giúp nhóm đạt mục tiêu Sprint?**
3. **Có trở ngại nào đang cản trở tôi hoặc cả nhóm hay không?**

#### 1.3 Yếu tố quan trọng để một Daily Scrum hiệu quả

- Bắt đầu đúng giờ – không chờ ai đến muộn.
- Tập trung vào **tiến độ Sprint Goal**, không lan man vào kỹ thuật.
- Trả lời đúng và đủ 3 câu hỏi.
- Scrum Master phải loại bỏ trở ngại sau buổi họp.
- Tất cả task phải được cập nhật trên Kanban trước cuộc họp.
- Không biến Daily Scrum thành buổi báo cáo cho Scrum Master.

## 2. Xây dựng Burndown Chart

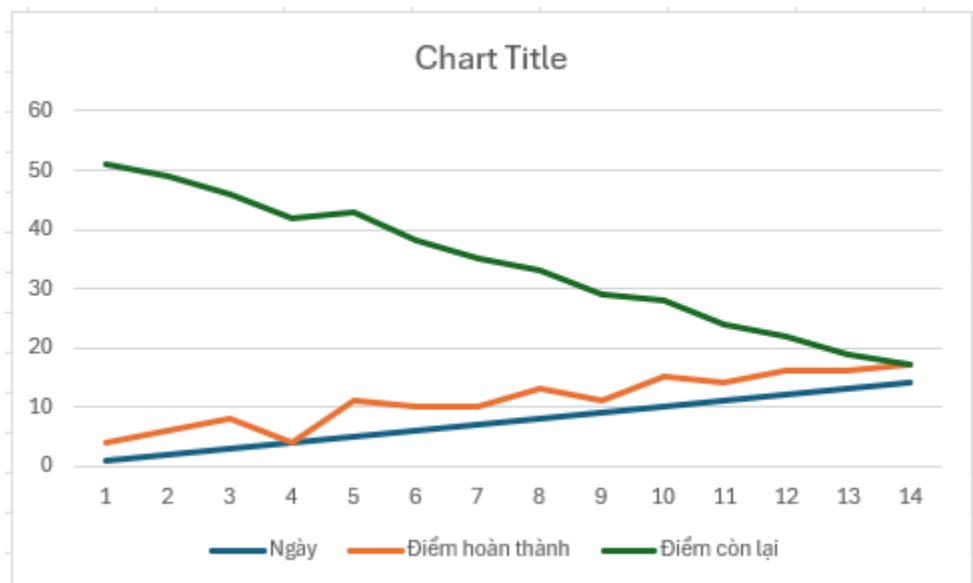
### 2.1 Dữ liệu Sprint

- Sprint: **2 tuần (14 ngày)**
- User Stories: **10 stories**
- Mỗi story: 3 → 8 điểm
- Tổng điểm Story = 50 điểm**

### 2.2 Bảng tiến độ (ví dụ mô phỏng 14 ngày)

Ngày	Điểm hoàn thành	Điểm còn lại
1	3	47
2	4	43
3	5	38
4	0	38
5	6	32
6	4	28
7	3	25
8	5	20
9	2	18
10	5	13
11	3	10
12	4	6
13	3	3
14	3	0

### 2.3 Burndown Chart



## Biểu đồ gồm 2 đường:

- **Đường lý tưởng (Ideal Line):** giảm từ 50 → 0 đều trong 14 ngày.
- **Đường thực tế (Actual Line):** dựa trên bảng điểm còn lại mỗi ngày.

Biểu đồ cho thấy cách lượng công việc giảm xuống theo thời gian.

## 2.4 Cách đọc và phân tích Burndown Chart

- Nếu đường thực tế **nằm phía dưới** đường lý tưởng → Nhóm làm vượt tiến độ.
- Nếu đường thực tế **nằm phía trên** đường lý tưởng → Nguy cơ trễ Sprint, cần điều chỉnh.
- Nếu biểu đồ bị **nằm ngang nhiều ngày** → Nhóm bị tắc nghẽn (bottleneck), có impediment.
- Scrum Master dùng biểu đồ để:
  - Kiểm tra tốc độ hoàn thành
  - Nhận diện chậm trễ
  - Đề xuất thay đổi kế hoạch làm việc

## 3. Quản lý công việc bằng Kanban Board

### 3.1 Bảng Kanban Board

Kanban Board		
To Do	In Progress	Done
Thiết kế giao diện Login	Thiết kế giao diện Login	
Tạo API đăng nhập	API đăng nhập	
		Tạo cơ sở dữ liệu người dùng
		Kiểm tra bảo mật API
Tối ưu hiệu năng		
Thiết kế dashboard		
Kiểm thử tính năng login		

## 4. Vai trò của Scrum Master trong Sprint

### 4.1 Đảm bảo Daily Scrum diễn ra hiệu quả

Scrum Master cần:

- **Điều phối** để cuộc họp đúng 15 phút.
- **Nhắc nhở** các thành viên trả lời đúng 3 câu hỏi.
- **Đảm bảo** mọi người cập nhật task trước cuộc họp.
- **Loại bỏ** ngay trở ngại sau khi nghe báo cáo.
- **Không để** cuộc họp bị biến thành buổi báo cáo cá nhân.

## 4.2 Giải quyết vấn đề và xung đột trong nhóm

Scrum Master sẽ:

- Lắng nghe và nhận diện nguyên nhân xung đột.
- Tổ chức cuộc họp riêng (Conflict Resolution Meeting).
- Nhắc lại Sprint Goal để thống nhất mục tiêu.
- Trong vấn đề kỹ thuật → để Developer tự thống nhất, Scrum Master chỉ điều phối.
- Trong vấn đề giao tiếp → áp dụng mô hình "Non-Violent Communication".
- Nếu xung đột ảnh hưởng Sprint → báo Product Owner và điều chỉnh kế hoạch.

## 4.3 Ba kỹ năng quan trọng của Scrum Master

1. Facilitation Skill – Kỹ năng điều phối cuộc họp hiệu quả.
2. Communication & Conflict Resolution – Giao tiếp tốt, xử lý xung đột công bằng.
3. Coaching & Mentoring – Hướng dẫn team hiểu Scrum và tự cải thiện.

Ngoài ra còn có: Quản lý rủi ro, quan sát, quản lý thay đổi.

Bài 7:

Kanban board

To Do	In Progress	Done
Thiết kế giao diện form đăng ký		
Kiểm thử form đăng ký		
Kết nối API đăng ký		

Sau khi scrum buổi sáng

To Do	In Progress	Done
Thiết kế giao diện form đăng ký	Kết nối API đăng ký	
Kiểm thử form đăng ký		

Tổng điểm công việc

- Tổng story points: **9 điểm**
- Chia đều:
  - Thiết kế UI: 3 points
  - Kết nối API: 3 points
  - Kiểm thử: 3 points
  -

## Mô phỏng Daily Scrum buổi sáng

**Scrum Master:** bắt đầu cuộc họp, giới hạn 15 phút, nhắc team trả lời 3 câu hỏi chuẩn.

### Thành viên 1 (Frontend)

#### Hôm qua tôi đã làm gì?

- Chuẩn bị wireframe cho trang đăng ký.

#### Hôm nay tôi sẽ làm gì?

- Hoàn thiện giao diện form đăng ký (task: Thiết kế UI).

#### Tôi có gặp cản trở gì không?

- Chưa rõ yêu cầu về các trường bắt buộc (email, xác nhận mật khẩu...). Cần Product Owner xác nhận.

### Thành viên 2 (Backend)

#### Hôm qua tôi đã làm gì?

- Kiểm tra lại API đăng ký hiện có và database user.

#### Hôm nay tôi sẽ làm gì?

- Triển khai kết nối API đăng ký vào giao diện.

#### Tôi có gặp cản trở gì không?

- API trả về mã lỗi chưa đồng nhất. Cần trao đổi lại với team về format response.

### Thành viên 3 (Tester)

#### Hôm qua tôi đã làm gì?

- Chuẩn bị test case cho form đăng ký.

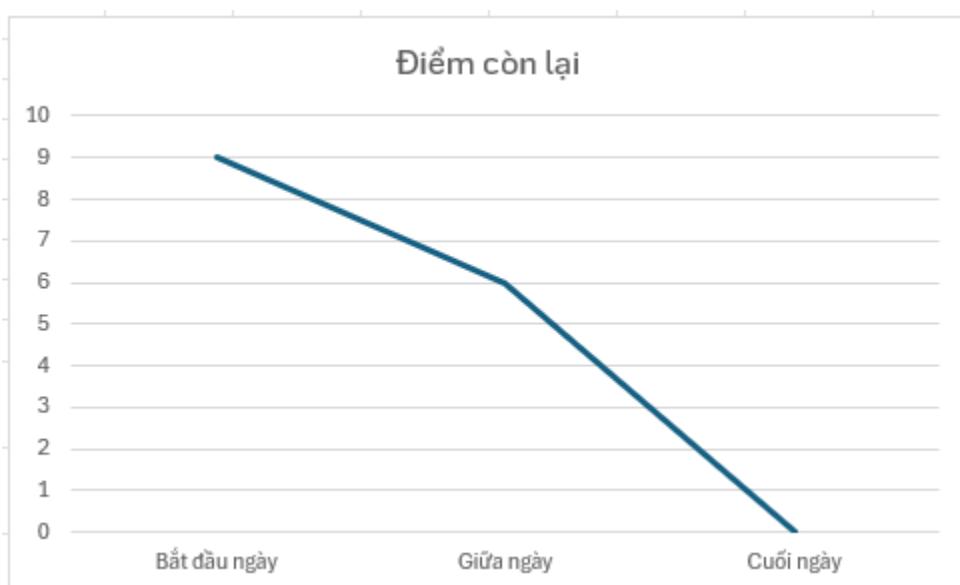
#### Hôm nay tôi sẽ làm gì?

- Kiểm thử form sau khi frontend + backend hoàn thiện.

#### Tôi có gặp cản trở gì không?

- Chưa có bản build ổn định để test.

### Burndown Chart cuối ngày



## Story Points còn lại theo thời gian (Sprint 1 ngày)

Thời điểm	Điểm còn lại
Bắt đầu ngày	9
Giữa ngày	6
Cuối ngày	0

→ Vì cả 3 task đều được hoàn thành trong ngày.

## Vai trò của Scrum Master trong ngày

Scrum Master thực hiện:

- **Tổ chức Daily Scrum** đúng 15 phút, đảm bảo mọi người tập trung vào tiến độ.
- **Gỡ blocker nhanh**: làm việc với Product Owner để xác nhận yêu cầu UI, hỗ trợ chuẩn hóa API response.
- **Theo dõi Kanban Board** để tránh quá tải, nhắc team pull task đúng thứ tự.
- **Quan sát tiến độ qua Burndown Chart** và thúc đẩy team hoàn thành mục tiêu trong ngày.
- **Đảm bảo giao tiếp mượt**, tránh gián đoạn hoặc lệch hướng.