

Ex1: Lên kịch bản Daily Scrum

- **Thời gian họp:** 8h00 – 8h15
- **Địa điểm họp:** Khu làm việc dự án

Các thành viên tham gia

Team	Thành viên
Dev	Thân Tùng
Design	Khánh Huyền
QA	Đạt
PO	Hưng

Bắt đầu cuộc họp (8h00)

Scrum Master (bạn) mở đầu:

- “Chào mọi người, bắt đầu Daily Scrum. Thời gian tối đa 15 phút.”
- “Mỗi người cập nhật: *Hôm qua làm gì – Hôm nay làm gì – Có cản trở gì – Giải pháp/đề xuất.*”
- “Vấn đề cần trao đổi sâu sẽ để ‘parking lot’ sau họp để không vượt thời gian.”

Nội dung cập nhật tiến độ

Tên thành viên	Hôm qua đã làm	Hôm nay sẽ làm	Cản trở, khó khăn	Giải pháp
Ngọc	Làm giao diện trang đăng nhập	Tích hợp API và lưu thông tin người dùng vào database	Chưa tìm hiểu xong thư viện UI (UI kit) nên làm UI chậm	Cho thêm 2 tiếng để tìm hiểu thư viện UI, sau đó cập nhật lại tiến độ và tiếp tục tích hợp

Thân Tùng (Dev)	Tạo API đăng nhập + kiểm tra tài khoản (validate)	Hoàn thiện API trả về token + chuẩn hóa format response để FE dễ xử lý	Chưa thống nhất format lỗi/response với phía giao diện	Chốt format response chung (success/error) ngay sau Daily (5–10 phút)
Khánh Huyền (Design)	Hoàn thiện layout Login + guideline màu/chữ	Thiết kế màn Home cơ bản + bàn giao file design	Chưa rõ yêu cầu hiển thị lỗi (error message) ở login	Làm nhanh 1 đề xuất UI lỗi + gửi Dev/PO duyệt trong sáng nay
Đạt (QA)	Viết test case cho Login (đúng/sai mật khẩu, validation)	Test tích hợp API login khi Dev build xong + log bug	Chưa có bản build ổn định để test end-to-end	Dev gửi bản build sau khi chốt API; QA test ngay và phản hồi trong ngày
Hưng (PO)	Xác nhận yêu cầu màn Login (trường dữ liệu, quy tắc)	Xác nhận yêu cầu luồng “Quên mật khẩu” + tiêu chí chấp nhận	Một số yêu cầu chưa rõ: lưu thông tin nào, có “Remember me” không	PO chốt yêu cầu chi tiết trong hôm nay để team làm đúng Sprint Goal

(Nếu đề bài bắt buộc chỉ ghi 2 dòng Ngọc, Linh thì bạn có thể đổi “Thân Tùng” thành “Linh” luôn. Nhưng theo danh sách team bạn đưa thì mình điền đúng role cho đủ.)

Kết thúc cuộc họp (8h15)

Scrum Master kết luận:

- “Cảm ơn mọi người. Hôm nay ưu tiên: **tích hợp login + chuẩn hóa response + QA test sớm.**”
- “Các vấn đề cần trao đổi thêm: Dev + Design chốt format lỗi và UI hiển thị lỗi (sau họp 10 phút).”
- “Kết thúc Daily, mọi người quay về vị trí và tiếp tục công việc.”

Bài 2:

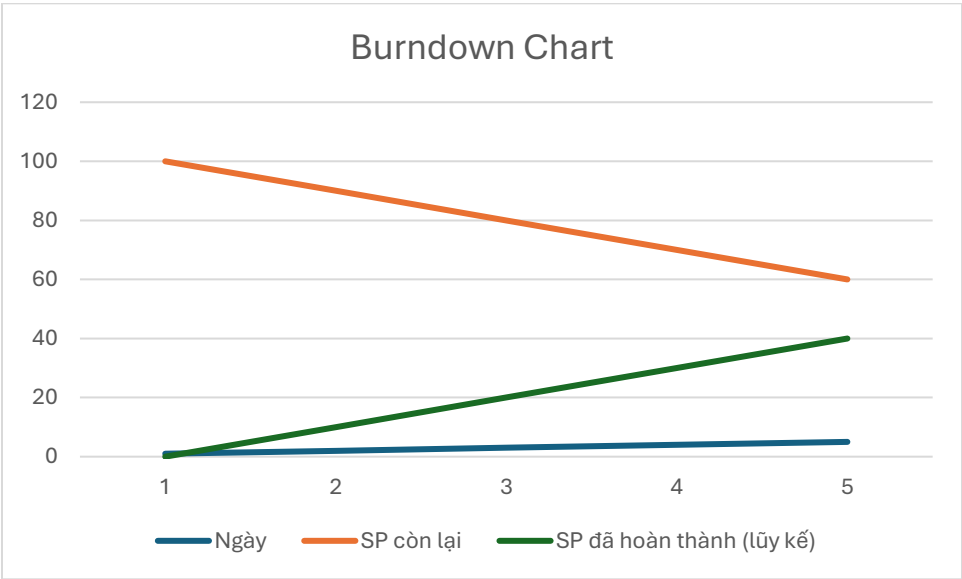
Bước 1) Tạo Burndown Chart đơn giản

- **Trục X:** Ngày trong Sprint (Day 1 → Day 10)
- **Trục Y:** Khối lượng công việc **còn lại** (story points)
- **Tổng khối lượng ban đầu:** 100 SP
- **Đường “Ideal” (lý tưởng):** giảm đều mỗi ngày
 - Burn lý tưởng: $100 / 10 = 10 \text{ SP/ngày}$

Bước 2) Mô phỏng 5 ngày đầu & cập nhật Burndown

Bạn cho dữ liệu “còn lại” như sau:

Ngày	SP còn lại	SP đã hoàn thành (lũy kế)
1	100	0
2	90	10
3	80	20
4	70	30
5	60	40



Cách chấm điểm lên biểu đồ: đặt các điểm (1,100), (2,90), (3,80), (4,70), (5,60) rồi nối lại thành “Actual line”.

Bước 3) Giải thích Burndown Chart: Nhóm có đi đúng hướng không?

Với dữ liệu trên, đường “Actual” đang **giảm đều 10 SP/ngày** → **đi đúng hướng** theo kế hoạch (bám sát tốc độ lý tưởng).

Ý nghĩa đọc nhanh:

- Đường giảm đều: tiến độ ổn định, ít “kẹt” story.
- Nếu có ngày phẳng (không giảm): thường do story quá lớn/đang chờ review/test/blocked.
- Nếu giảm rất mạnh ở cuối: dấu hiệu “dồn việc”, đóng story muộn.

Hành động Scrum Master nên làm để giữ nhịp:

- Nhắc team bám **DoD (Definition of Done)** để story “Done” thật sự.
 - Giảm **WIP** (đang làm dở quá nhiều) → ưu tiên “finish > start”.
 - Gỡ blocker nhanh (phối hợp PO/BA/DevOps/QA nếu cần).
-

Bước 4) Nếu Burndown không giảm đều (ví dụ Ngày 5 chỉ còn “hoàn thành 40” thay vì “60”)

Trường hợp hay gặp là: **đáng lẽ đến ngày 5 team phải hoàn thành ~60 SP (còn 40)**, nhưng thực tế **mới hoàn thành 40 SP (còn 60) ⇒ trễ 20 SP**.

4.1. Phân tích nguyên nhân (thường gặp)

- Story **quá to**, chưa “Done” nên burndown đứng/giảm ít.
- Thiếu/đổi yêu cầu, phát sinh bug, phụ thuộc API/back-end.
- Test/QA/review bị dồn về cuối, thiếu automation/CI.
- Ước lượng sai hoặc capacity thực tế thấp hơn dự kiến.

4.2. Hành động điều chỉnh (cụ thể, làm ngay)

1. **Swarm để đóng story đang dở**

- Ưu tiên kéo story “In Progress” → “Done”, hạn chế mở story mới.

2. Chẻ nhỏ story

- Tách thành phần có thể bàn giao độc lập (UI/API/validation/report...).

3. Rà soát blocker trong Daily Scrum

- Ai blocked bởi gì? cần ai hỗ trợ? chốt thời hạn gỡ blocker trong ngày.

4. Làm việc với PO về phạm vi

- Nếu vẫn trể: **giảm scope** (đẩy bớt item ít ưu tiên ra khỏi Sprint) để bảo vệ Sprint Goal.

5. Tăng tốc chất lượng

- Đẩy code review nhanh, test sớm, thêm checklist DoD, ưu tiên fix bug ảnh hưởng tiến độ.

Kết quả mong muốn: burndown quay lại “đi xuống đều” trong vài ngày tiếp theo, hoặc điều chỉnh scope để vẫn đạt **Sprint Goal** đúng hạn.

Bài 3:

1. Bảng mô phỏng 1 Daily Scrum (ngày cụ thể trong Sprint)

Thành viên	Hôm qua đã làm gì?	Hôm nay sẽ làm gì?	Blocker/Vướng mắc
An (Front-end)	Xong UI Login + validate form, gọi API mock	Tích hợp API thật login, lưu token, nối màn Services	Response lỗi khác spec
Bình (Back-end)	Xong /auth/login, /services, deploy staging	Chuẩn hóa response + cập nhật Swagger	Thiếu rule “sai 5 lần” xử lý sao
Dũng (Mobile/Integration)	Tích hợp /services nhưng lỗi môi trường	Fix baseUrl/SSL, test end-to-end	Không có tài khoản test/OTP bypass

Chi (QA)	Viết test cases login/services	Test end-to-end trên staging, log bug P0	AC lock account/OTP retry chưa rõ
----------	--------------------------------	--	-----------------------------------

2. Bảng mô phỏng cách nhóm xử lý vấn đề ngay trong cuộc họp

Vấn đề	Ảnh hưởng	Cách xử lý ngay trong Daily (chốt nhanh)	Người phụ trách
FE-BE khó tích hợp (contract lệch)	FE parse lỗi sai, QA khó verify	BE chuẩn hóa response + update Swagger, FE tích hợp theo Swagger mới	Bình + An
Thiếu tài liệu/yêu cầu từ PO/khách hàng	Không test/Dev đúng tiêu chí	Parking lot 10 phút sau Daily để chốt AC và cập nhật user story	SM + Bình + Chi

3. Bảng lời khuyên để Daily Scrum ngắn gọn & hiệu quả

Nguyên tắc	Cách áp dụng	Mục tiêu
Timebox 15 phút	Đặt timer, mỗi người 60–90 giây	Không kéo dài
Chỉ 3 câu hỏi	Hôm qua / Hôm nay / Blocker	Không lan man
Bám Sprint Goal	Nhắc mục tiêu 24h ngay đầu họp	Đúng trọng tâm
Parking Lot	Vấn đề cần tranh luận → ghi lại họp sau	Daily không biến thành “technical meeting”
Update board ngay	Kéo task đúng trạng thái (In progress/Review/Done)	Minh bạch tiến độ
Giảm WIP	Ưu tiên “finish > start”	Tránh dồn việc cuối Sprint

4. Bảng mô phỏng vai trò Scrum Master trong cuộc họp (SM làm gì?)

Scrum Master làm gì?	Nói/Thực hiện như thế nào?	Kết quả đạt được
----------------------	----------------------------	------------------

Mở đầu đúng trọng tâm	“Mục tiêu hôm nay: chạy end-to-end login → services trên staging.”	Team cùng 1 hướng
Điều phối thời gian	“Ý này để parking lot sau daily nhé.”	Giữ 10–15 phút
Làm rõ blocker	“Blocked bởi gì? cần ai? bao giờ gỡ xong?”	Blocker thành việc cụ thể
Chốt action items	Ghi việc + owner + deadline trong ngày	Có kế hoạch xử lý
Bảo vệ team	SM chủ động liên hệ PO/DevOps/khách hàng để gỡ phụ thuộc	Team không bị đứng
Kết thúc rõ ràng	Tóm tắt 2–3 việc quan trọng + ai làm	Không mở hồ sau họp

Bài 4:

1. Tạo Burndown Chart cho Sprint 10 ngày (Tổng 120 SP)

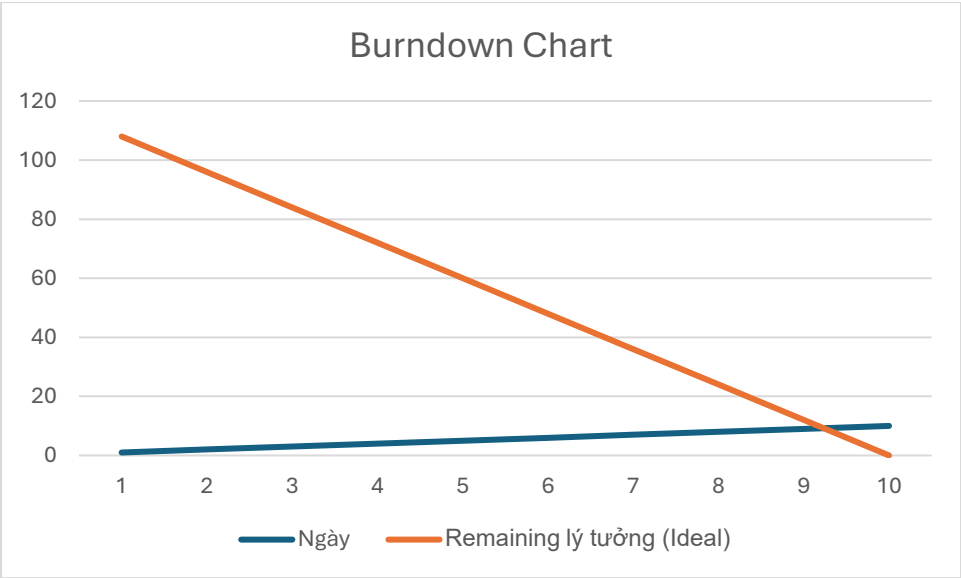
Quy ước:

- Trục X: Ngày 1 → Ngày 10
- Trục Y: **Story points còn lại (Remaining)** vào **cuối mỗi ngày**
- **Ideal burn:** $120 / 10 = 12 \text{ SP/ngày}$

Bảng dữ liệu nền (Ideal line):

Ngày	Remaining lý tưởng (Ideal)
1	108
2	96
3	84
4	72
5	60

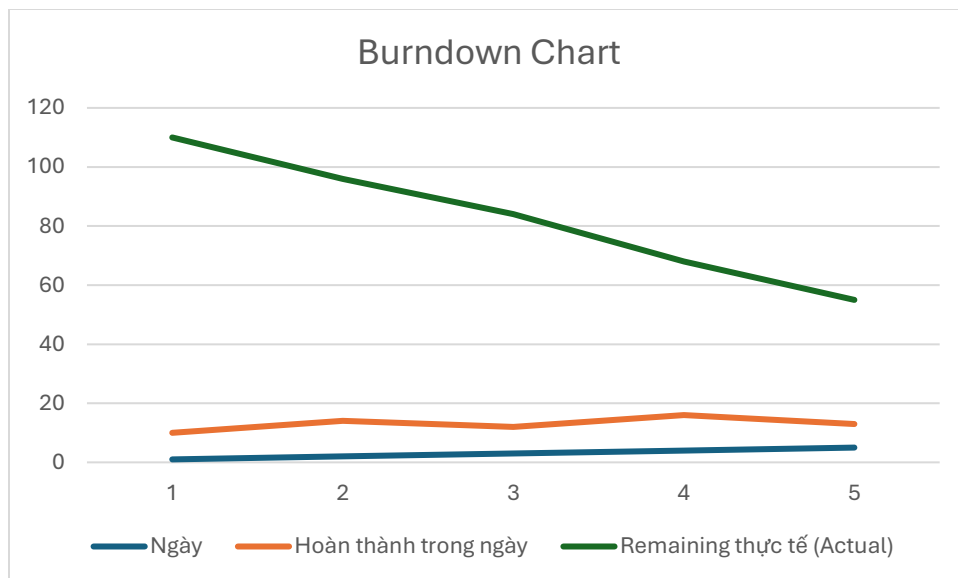
6	48
7	36
8	24
9	12
10	0



2. Mô phỏng cập nhật Burndown Chart sau 5 ngày (có SP hoàn thành mỗi ngày)

Giả sử team cập nhật như sau (cuối ngày):

Ngày	Hoàn thành trong ngày	Remaining thực tế (Actual)
1	10	110
2	14	96
3	12	84
4	16	68
5	13	55



Nhìn vào đường cong (Actual) 5 ngày đầu:

- Remaining giảm khá đều và **đến ngày 5 còn 55**, trong khi **Ideal ngày 5 là 60**
→ Team đang **nhANH hơn kế hoạch ~5 điểm** (tạm thời “on track”, thậm chí hơi ahead).

3. Ngày 6 gặp vấn đề tài nguyên → Burndown phản ánh thế nào & cách điều chỉnh

Tình huống: Ngày 6 thiếu người/thiếu tài nguyên → tiến độ chậm.

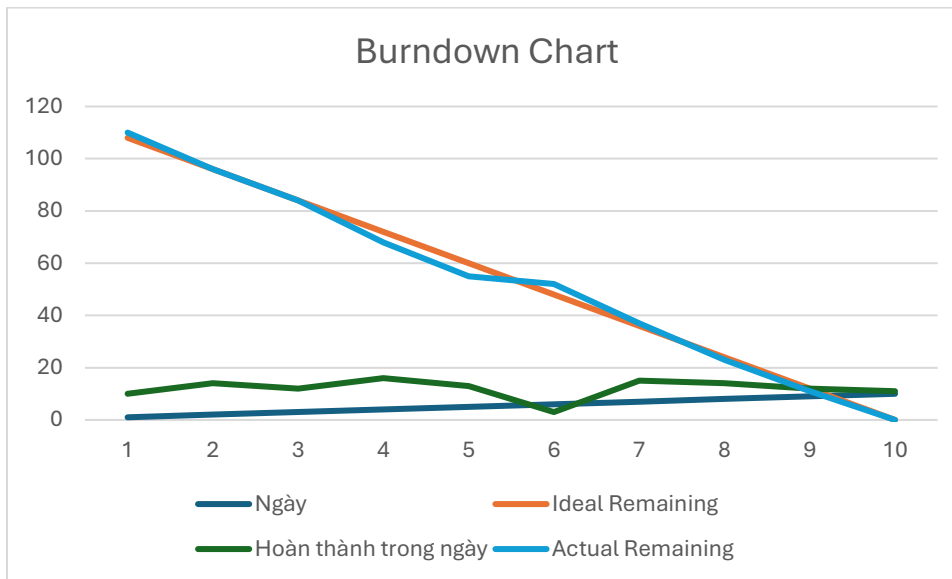
Giả sử:

- Ngày 6 chỉ hoàn thành **3 SP** ⇒ Remaining **52** (đáng lẽ lý tưởng là 48)

Cập nhật tiếp đến hết Sprint (có điều chỉnh để kịp Sprint Goal):

Ngày	Ideal Remaining	Hoàn thành trong ngày	Actual Remaining
1	108	10	110
2	96	14	96
3	84	12	84
4	72	16	68
5	60	13	55
6	48	3	52

7	36	15	37
8	24	14	23
9	12	12	11
10	0	11	0



Cách Burndown “nói lên vấn đề”:

- Ở ngày 6, đường Actual giảm ít hơn bình thường \Rightarrow dấu hiệu **chậm tiến độ** (Actual 52 > Ideal 48).
- Sau đó team **tăng tốc** ngày 7–10 để kéo về 0.

Team điều chỉnh để khắc phục (ngắn gọn, đúng Scrum):

- **Swarm**: tập trung đóng các story gần Done (Finish > Start), giảm WIP.
- **Chẻ nhỏ story** để “Done” sớm, tránh dồn đến cuối.
- **Gỡ blocker** (phụ thuộc API/QA/DevOps) bằng cuộc họp nhỏ sau Daily (“parking lot”).
- Nếu không thể tăng tốc đủ: **đàm phán với PO** cắt bớt hạng mục ít ưu tiên để bảo vệ Sprint Goal.

4. Vai trò Scrum Master khi dùng Burndown Chart (khi đường cong không giảm đều)

Scrum Master cần làm các việc sau để phát hiện sớm và xử lý đúng:

1. Theo dõi Burndown mỗi ngày và “đọc” tín hiệu:

- Đường **phẳng/giảm rất ít**: story quá to, review/test bị nghẽn, blocked, WIP cao
- Đường **tụt mạnh cuối Sprint**: dấu hiệu dồn việc, “Done” muộn, rủi ro không kịp

2. Khi phát hiện lệch khỏi ideal:

- Hỏi nhanh trong Daily: **Blocked bởi gì? cần ai? khi nào gỡ xong?**
- Facilitate để team chốt **action items** (owner + deadline trong ngày)

3. Bảo vệ tiến độ & Sprint Goal:

- Gỡ impediments ngoài team (PO/khách hàng/DevOps)
- Nhắc team bám **Definition of Done** để story “Done” thật
- Nếu cần: hỗ trợ PO **điều chỉnh scope** (loại bớt item ít ưu tiên)

Bài 5:

1. Xây dựng Kanban Board (To Do / In Progress / Done) + Cards trong Sprint

1.1. Danh sách thẻ công việc (Cards) của Sprint (ví dụ ứng dụng quản lý công việc)

Mã	Card (Công việc)	Mô tả ngắn	Người phụ trách	Ưu tiên
C1	Thiết kế UI Dashboard	Wireframe + UI danh sách task	UI/UX	High
C2	Thiết kế UI Login/Register	Form + validate	UI/UX	High
C3	Phát triển API Auth	Login, Register, JWT	Back-end	High
C4	Phát triển API Task CRUD	Create/Update/Delete/List	Back-end	High

C5	Tích hợp Frontend với API Auth	Gọi API + lưu token	Front-end	High
C6	Tích hợp Frontend với API Task	Hiển thị list + thao tác CRUD	Front-end	High
C7	Kiểm thử bảo mật cơ bản	OWASP basic, kiểm tra JWT	QA/Security	Medium
C8	Viết Unit Test API	Test auth + task endpoint	Back-end	Medium
C9	Kiểm thử chức năng (E2E)	Luồng login → tạo task → sửa → xóa	QA	High
C10	Sửa bug & Refactor	Fix bug từ QA	Dev	Medium

1.2. Kanban Board (trạng thái ban đầu)

To Do	In Progress	Done
C1, C2, C3, C4, C5, C6, C7, C8, C9, C10	(trống)	(trống)

2. Đặt giới hạn (WIP Limit) cho mỗi cột + giải thích

2.1. WIP Limit đề xuất

Cột	WIP Limit	Lý do
To Do	Không giới hạn (hoặc tối đa 10–15)	Đây là backlog của Sprint, chỉ để “chờ làm”
In Progress	Tối đa 3 cards	Tránh làm quá nhiều việc cùng lúc, giảm multi-tasking, tăng tốc “Done”
Done	Không giới hạn	Thể hiện kết quả hoàn thành

2.2. Vì sao WIP Limit quan trọng với nhóm Scrum?

- Giúp team **tập trung hoàn thành** thay vì mở quá nhiều việc.
- Giảm tình trạng “**kẹt ở In Progress**” (dev xong nhưng chưa test/review).
- Lộ rõ **nút thắt cổ chai** (bottleneck) như thiếu QA/review chậm.
- Tăng **flow** → thẻ đi qua board nhanh hơn → dự đoán tiến độ tốt hơn.

3. Mô phỏng quá tải ở “In Progress” và cách tối ưu lại quy trình

3.1. Tình huống quá tải

Giả sử team “lỡ kéo” quá nhiều cards vào In Progress:

To Do	In Progress (bị quá tải)	Done
C7, C8, C9, C10	C1, C2, C3, C4, C5, C6 (6 cards)	(trống)

Vấn đề nhìn thấy ngay:

- In Progress vượt WIP ($6 > 3$) \Rightarrow team đang **đa nhiệm**, dễ kéo dài, khó hoàn thành.

3.2. Điều chỉnh Kanban Board để giải quyết

(A) Thay đổi cấu trúc cột để thấy bottleneck rõ hơn

Thay vì 3 cột, tối ưu thành 5 cột:

To Do	In Dev	Code Review	Testing	Done
-------	--------	-------------	---------	------

(B) Đặt WIP Limit theo từng giai đoạn

Cột	WIP Limit gợi ý
In Dev	3
Code Review	2
Testing	2

(C) Quy tắc vận hành mới (policy)

- Nếu **Testing** đầy \rightarrow dev **dừng kéo việc mới**, tập trung hỗ trợ QA, fix bug để “thoát tắc”.
- Nếu **Code Review** đầy \rightarrow ưu tiên review trước, không mở thêm task mới.
- Áp dụng “**Swarm**”: 2–3 người cùng xử lý để đẩy 1 card tới Done nhanh.

3.3. Sau khi tối ưu (ví dụ trạng thái hợp lý)

To Do	In Dev (≤ 3)	Code Review (≤ 2)	Testing (≤ 2)	Done
C7, C8, C10	C3, C4, C5	C1	C2	(trống)

4. Vai trò Scrum Master khi dùng & tối ưu Kanban Board

Scrum Master cần làm	Cách thực hiện	Mục tiêu
Thiết lập board rõ ràng	Định nghĩa cột + policy (khi nào kéo thẻ, khi nào xong)	Team hiểu và dùng thống nhất
Đặt & giữ WIP limit	Nhắc team không vượt WIP, nếu vượt thì xử lý tắc nghẽn trước	Tránh quá tải, tăng tốc độ hoàn thành
Phát hiện bottleneck	Nhìn cột nào “tắc” (Review/Testing) và tổ chức swarm	Giải quyết kẹt việc sớm
Facilitate cải tiến liên tục	Retro: phân tích thẻ stuck, lead time, nguyên nhân	Board ngày càng hiệu quả
Tăng phối hợp	Khuyến khích pair work, review nhanh, hỗ trợ chéo	Team flow tốt hơn
Bảo vệ team	Gỡ impediments ngoài team (môi trường test, quyền truy cập, PO chậm phản hồi)	Không bị đứng việc

Bài 6:

1) Kế hoạch Daily Scrum

1.1. Cách tổ chức Daily Scrum chuẩn (kế hoạch)

Mục	Nội dung
Thời gian	9:00–9:15 mỗi ngày (timebox 15 phút), cố định giờ
Thành phần	Dev Team + Scrum Master (PO có thể tham dự nhưng không điều khiển)
Hình thức	Đứng họp nhanh, mở board (Jira/Trello/Excel) để nhìn cùng 1 nguồn
Mục tiêu	Đồng bộ tiến độ 24h tới + phát hiện blocker sớm để gỡ ngay
Quy tắc	Không giải quyết tranh luận kỹ thuật trong Daily (đưa vào “Parking lot” sau họp)

1.2. 3 câu hỏi chính trong Daily Scrum

#	Câu hỏi
1	Hôm qua bạn đã làm gì để giúp nhóm đạt Sprint Goal?
2	Hôm nay bạn sẽ làm gì để giúp nhóm đạt Sprint Goal?
3	Bạn có gặp vướng mắc/blocked gì không? (cần ai hỗ trợ?)

1.3. Yếu tố quan trọng để Daily Scrum hiệu quả

Yếu tố	Cách làm
Timebox	15 phút, mỗi người 60–90 giây
Bám Sprint Goal	Nhắc “mục tiêu 24h tới” ngay đầu cuộc họp
Minh bạch	Update trạng thái thẻ ngay trên board (To Do/In Progress/Review/Done)
Tập trung gỡ blocker	Blocker phải ra được “owner + deadline”
Parking lot	Vấn đề cần bàn sâu → họp riêng 5–10 phút sau Daily với đúng người

2) Burndown Chart (10 User Stories, tổng 50 điểm)

2.1. Danh sách 10 User Stories (3–8 điểm, tổng = 50)

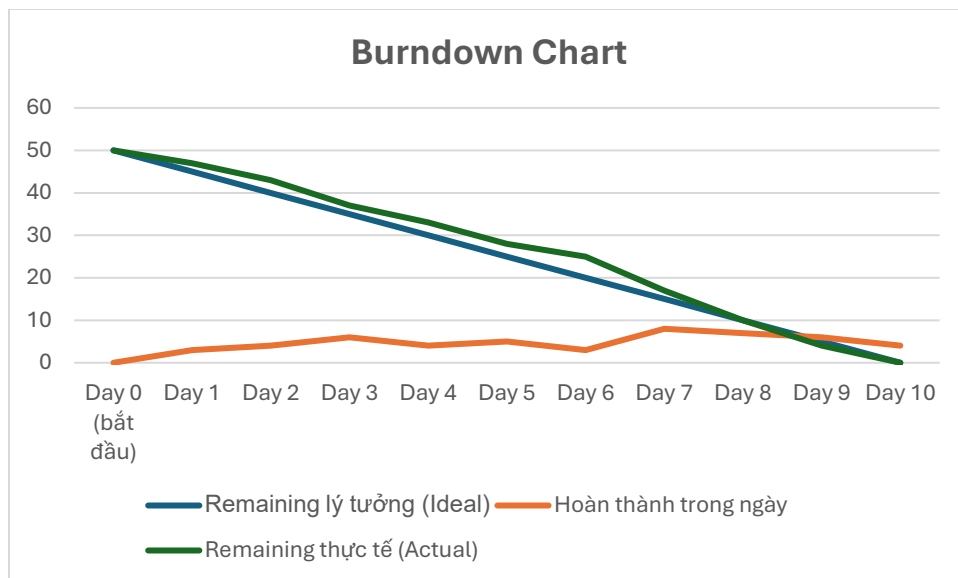
US	User Story	Story Points
US1	Đăng nhập/Đăng xuất	5
US2	Tạo task mới	8
US3	Sửa task	5
US4	Xóa task	3
US5	Danh sách task + lọc (status/priority)	8
US6	Gán task cho thành viên	5
US7	Due date + nhắc hạn	5
US8	Bình luận trong task	3

US9	Nhật ký hoạt động (activity log)	3
US10	Dashboard thống kê cơ bản	5
	Tổng	50

2.2. Bảng cập nhật Burndown theo ngày (10 ngày)

Ideal rate: 50 điểm / 10 ngày = **5 điểm/ngày**

Ngày	Remaining lý tưởng (Ideal)	Hoàn thành trong ngày	Remaining thực tế (Actual)
Day 0 (bắt đầu)	50	0	50
Day 1	45	3	47
Day 2	40	4	43
Day 3	35	6	37
Day 4	30	4	33
Day 5	25	5	28
Day 6	20	3	25
Day 7	15	8	17
Day 8	10	7	10
Day 9	5	6	4
Day 10	0	4	0



Cách vẽ nhanh: Trục X = Day 0→10, Trục Y = Remaining. Vẽ 2 đường: **Ideal** và **Actual**.

2.3. Cách đọc & phân tích Burndown Chart

Dấu hiệu trên Burndown	Ý nghĩa	Bạn nên làm gì (Scrum Master)
Actual > Ideal nhiều ngày	Đang chậm tiến độ	Tìm bottleneck (review/test/blocked), giảm WIP, swarm để “Done”
Đường Actual bị “phẳng”	Task to/đợi review/đợi test	Chẻ nhỏ story, ưu tiên đóng việc đang làm dở
Tụt mạnh cuối Sprint	Dồn “Done” cuối kỳ, rủi ro cao	Đẩy test/review sớm, nhắc Definition of Done
Actual bám sát/nhỏ hơn Ideal	On track hoặc ahead	Giữ nhịp, tránh kéo thêm scope ngoài Sprint

3) Quản lý công việc bằng Kanban Board (To Do / In Progress / Code Review / Done)

3.1. Kanban Board (ví dụ snapshot giữa Sprint – cuối Day 4)

To Do	In Progress	Code Review	Done
US6, US7, US8, US9, US10	US3, US5	US2	US1, US4

3.2. Cách dùng Kanban Board để theo dõi & quản lý trong Sprint

Cách sử dụng	Mục đích
Pull system (kéo việc)	Chỉ kéo việc mới khi cột sau không bị tắc
Thiết lập WIP (khuyến nghị)	Ví dụ: In Progress ≤ 3 , Code Review ≤ 2 để tránh quá tải
Định nghĩa rõ “xong ở mỗi cột”	In Progress: code xong + unit test; Code Review: đã review/approve; Done: đạt DoD
Theo dõi thẻ bị kẹt	Thẻ đứng lâu ở Review/Testing = bottleneck cần xử lý
Đồng bộ với Daily	Daily nhìn board để nói đúng tiến độ + blocker

4) Vai trò Scrum Master

4.1. Scrum Master làm gì để Daily Scrum hiệu quả?

Việc làm	Cách thực hiện
Giữ đúng timebox & đúng format	Nhắc 3 câu hỏi, cắt lan man → đưa “Parking lot”
Tập trung vào Sprint Goal	Mở đầu 20–30s: mục tiêu 24h tới
Biến blocker thành hành động	“Blocked bởi gì? cần ai? deadline?” → ghi action item rõ ràng
Đảm bảo minh bạch trên board	Nhắc update thẻ ngay sau khi báo cáo

4.2. Giải quyết vấn đề & xung đột trong suốt Sprint (cách làm thực tế)

Tình huống	Cách Scrum Master xử lý
Mâu thuẫn dev–dev / dev–QA	Gặp 1-1 để hiểu nguyên nhân → họp ngắn với 2 bên, thống nhất “working agreement”
Blocked do phụ thuộc (PO/DevOps/API)	SM chủ động liên hệ đúng người để gỡ, không để team tự bơi
Tranh luận kỹ thuật dài	Chuyển sang parking lot / technical huddle sau Daily, chỉ mời người liên quan

Scope thay đổi giữa Sprint	Làm việc với PO: ưu tiên lại, cắt/đẩy bớt item để bảo vệ Sprint Goal
----------------------------	--

4.3. Ít nhất 3 kỹ năng quan trọng của Scrum Master

Kỹ năng	Vì sao quan trọng
Facilitation (điều phối)	Giữ sự kiện Scrum ngắn gọn, đúng mục tiêu
Communication (giao tiếp)	Truyền đạt rõ ràng, kết nối Dev-PO-Stakeholder
Impediment removal (gỡ trở ngại)	Giúp team không bị đứng việc, tăng flow
Conflict resolution (giải quyết xung đột)	Giữ môi trường làm việc tích cực, hợp tác
Coaching/Servant leadership	Dẫn dắt bằng hỗ trợ, giúp team tự tổ chức tốt hơn

Bài 7:

Bước 1) Kanban Board (To Do – In Progress – Done)

To Do	In Progress	Done
T1. Thiết kế UI form đăng ký (3đ)		
T2. Kết nối API đăng ký (3đ)		
T3. Kiểm thử form đăng ký (3đ)		

Mô phỏng luồng trong ngày (gợi ý cập nhật theo mốc):

- **09:00:** T1 → In Progress
- **11:00:** T1 → Done, T2 → In Progress
- **15:00:** T2 → Done, T3 → In Progress
- **17:30:** T3 → Done

Bước 2) Tổng điểm công việc

- Tổng: **9 story points**
 - Mỗi task: **3 points**
 - **T1** (UI) = 3
 - **T2** (API) = 3
 - **T3** (Test) = 3
-

Bước 3) Daily Scrum buổi sáng (3 câu hỏi chuẩn, 2–3 dòng/người)

Thành viên:

- Dev A (Front-end)
- Dev B (Back-end)
- Scrum Master (bạn) *(có thể kiêm dev hoặc không)*

Dev A (Front-end)

- Hôm qua: (Sprint 1 ngày, coi như “chuẩn bị”) Em đã phác thảo layout form + kiểm tra yêu cầu field (name, email, password, confirm).
- Hôm nay: Em sẽ hoàn thiện UI form + validate (required, email format, password match), bàn giao cho phần tích hợp API.
- Blocker: Chưa rõ format lỗi trả về từ API (message/code) để hiển thị đúng trên UI.

Dev B (Back-end)

- Hôm qua: Chuẩn bị endpoint /register, validate input cơ bản, dự tính response mẫu.
- Hôm nay: Hoàn thiện API đăng ký + trả response chuẩn, hỗ trợ FE test bằng Postman.
- Blocker: Chưa có rule rõ cho password (độ dài, ký tự đặc biệt) và xử lý trùng email.

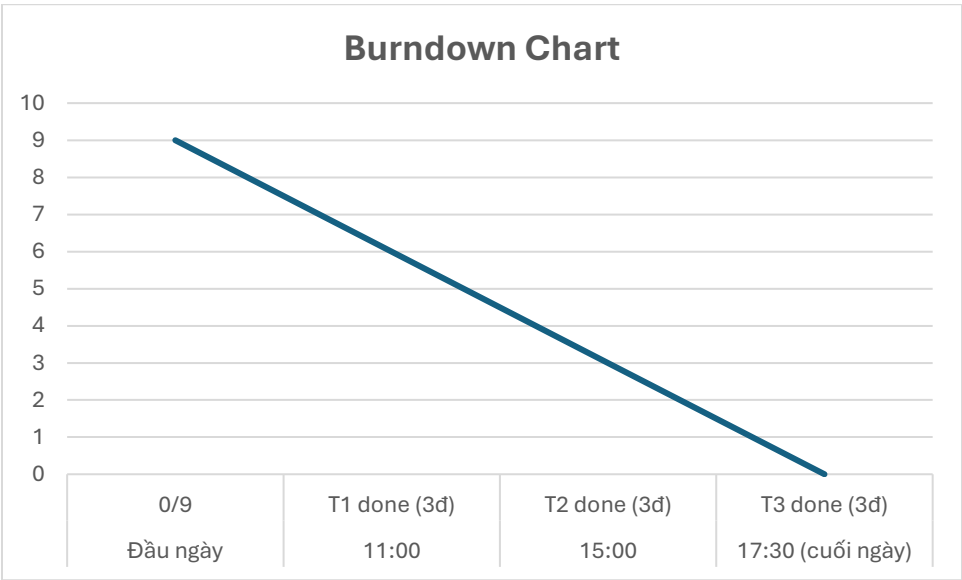
Scrum Master

- Hôm qua: Chốt 3 task, phân công, chuẩn bị board + tiêu chí Done.
- Hôm nay: Giữ WIP = 1 task/lần, ưu tiên “Done” từng task; gỡ blocker spec/error format với PO/nhóm.
- Blocker: Rủi ro dồn test cuối ngày nếu API trễ.

Bước 4) Cập nhật Burndown Chart cuối ngày (9 → 0)

Bảng dữ liệu (Remaining story points)

Thời điểm	Công việc Done	Remaining
Đầu ngày	0/9	9
11:00	T1 done (3đ)	6
15:00	T2 done (3đ)	3
17:30 (cuối ngày)	T3 done (3đ)	0



Đường Burndown: giảm 9 → 6 → 3 → 0.

Bước 5) Vai trò Scrum Master (ngắn gọn)

Scrum Master đảm bảo nhóm bám mục tiêu trong 1 ngày bằng cách giữ **Kanban minh bạch**, giới hạn **WIP** để tránh làm dờ nhiều việc, và phát hiện **blocker** sớm trong Daily Scrum. Khi thấy rủi ro trễ (ví dụ API/spec chưa rõ), Scrum Master chủ động **gỡ vướng** (chốt rule/format response), điều phối team “swarm” để hoàn thành theo thứ tự ưu tiên và đảm bảo “Done” đúng tiêu chí.

Bài 8:

1. Kanban Board (To Do – In Progress – Done)

Kanban Board đầu ngày

To Do	In Progress	Done
T1. Thiết kế UI form đăng ký		
T2. Kết nối API đăng ký		
T3. Kiểm thử form đăng ký		

Cập nhật trạng thái theo mốc trong ngày (mô phỏng)

Thời điểm	To Do	In Progress	Done
09:00	T2, T3	T1	
11:00	T3	T2	T1
15:00		T3	T1, T2
17:30			T1, T2, T3

2. Tổng điểm công việc (Story Points)

Giả sử tổng **9 story points**, mỗi task **3 điểm**:

Task	Điểm
T1. Thiết kế UI form đăng ký	3
T2. Kết nối API đăng ký	3
T3. Kiểm thử form đăng ký	3
Tổng	9

3. Daily Scrum buổi sáng (3 câu hỏi chuẩn, 2–3 dòng/người)

Thành viên nhóm (3 người):

- Dev A (Front-end)
- Dev B (Back-end)

- Scrum Master (SM)

Dev A (Front-end)

- Hôm qua: Chuẩn bị wireframe form, xác định field: họ tên, email, mật khẩu, confirm.
- Hôm nay: Làm UI + validate (required, email format, password match), bàn giao để tích hợp API.
- Blocker: Cần biết format lỗi API (code/message) để hiển thị thông báo đúng.

Dev B (Back-end)

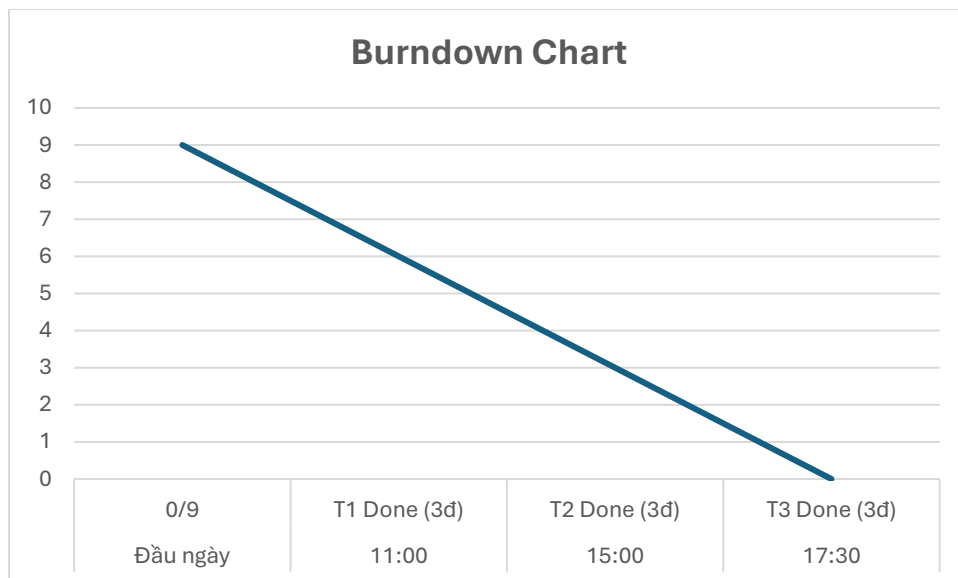
- Hôm qua: Chuẩn bị endpoint đăng ký /register, validate input cơ bản.
- Hôm nay: Hoàn thiện API + response chuẩn, hỗ trợ test bằng Postman.
- Blocker: Chưa chốt rule mật khẩu (độ dài/tối thiểu ký tự) và xử lý trùng email.

Scrum Master (SM)

- Hôm qua: Chốt phạm vi Sprint 1 ngày, tạo Kanban Board, thống nhất “Done”.
- Hôm nay: Giữ WIP thấp (chỉ 1 task In Progress), gỡ blocker yêu cầu/format lỗi sớm để không trễ.
- Blocker: Rủi ro dồn kiểm thử cuối ngày nếu API hoàn thành muộn.

4. Burndown Chart cuối ngày (9 → 0)

Thời điểm	Việc hoàn thành	Remaining (điểm còn lại)
Đầu ngày	0/9	9
11:00	T1 Done (3đ)	6
15:00	T2 Done (3đ)	3
17:30	T3 Done (3đ)	0



Dữ liệu để vẽ đường Burndown: 9 → 6 → 3 → 0

5. Vai trò Scrum Master (ngắn gọn)

Scrum Master đảm bảo tiến độ trong ngày bằng cách giữ Kanban Board minh bạch, nhắc team làm theo luồng “Done từng task”, và giới hạn WIP để tránh quá tải. Khi phát hiện rủi ro (thiếu rule mật khẩu/format lỗi API), Scrum Master chủ động phối hợp chốt yêu cầu nhanh và gỡ blocker, giúp nhóm hoàn thành đủ 3 task và burndown về 0.

Bài 9:

1) Phân tích vấn đề (ít nhất 4 vấn đề lớn) — dấu hiệu, nguyên nhân, rủi ro

Vấn đề 1: Daily Scrum “lạc mục tiêu” (không bám Sprint Goal)

- **Dấu hiệu:** Dev chỉ nói “hôm qua/hôm nay làm gì” nhưng **không liên hệ Sprint Goal** (đặt lịch + nhắc lịch).
- **Nguyên nhân có thể:** team quen báo cáo theo task cá nhân; Sprint Goal chưa được nhắc lại hằng ngày; backlog chưa được chia theo dòng giá trị (booking flow/reminder flow).
- **Rủi ro:** làm xong nhiều việc “bận rộn” nhưng **không tạo ra luồng đặt lịch end-to-end**, đến cuối Sprint không đạt mục tiêu.

Vấn đề 2: Che giấu/không nói thật về trở ngại (impediments bị “ẩn”)

- **Dấu hiệu:** 2 thành viên **không nêu blocker** dù **velocity** ngày đó **thấp**; burndown gần như đi ngang.
- **Nguyên nhân có thể:** sợ bị đánh giá; chưa có “an toàn tâm lý”; dev không nhận ra “blocked” (đang chờ API, chờ review, chờ test) cũng là trở ngại.
- **Rủi ro:** trở ngại không được gỡ kịp thời → công việc kẹt ở In Progress, dồn cuối Sprint, chất lượng giảm.

Vấn đề 3: Daily Scrum kéo dài 22–25 phút (mất hiệu quả, biến thành họp giải quyết vấn đề)

- **Dấu hiệu:** vượt timebox 15 phút.
- **Nguyên nhân có thể:** đi sâu kỹ thuật/ tranh luận; không có “parking lot”; update không dựa vào board; số lượng việc đang làm dở quá nhiều nên mọi người giải thích dài.
- **Rủi ro:** tốn thời gian, mệt team, thông tin loãng; vấn đề vẫn không được xử lý đúng chỗ.

Vấn đề 4: Kanban tắc nghẽn nghiêm trọng (WIP quá cao)

- **Dấu hiệu ngày 5:** In Progress = 8, Done = 2, To Do = 12.
- **Nguyên nhân có thể:** không có WIP limit; team “start nhiều, finish ít”; story/task quá to; nghẽn ở review/test/QA.
- **Rủi ro:** lead time tăng, chất lượng giảm, nhiều việc “gần xong” nhưng không “Done”, burndown đi ngang.

Vấn đề 5: Burndown chart lệch chuẩn (gần như đi ngang: 52→47 sau 5 ngày)

- **Dấu hiệu:** 5 ngày chỉ burn được **5 points** ⇒ tốc độ cực thấp so với kỳ vọng.
- **Nguyên nhân có thể:** story không được đóng “Done” (do DoD cao, thiếu test/review); bị block; scope thay đổi; ước lượng sai hoặc capacity bị hụt.
- **Rủi ro:** nếu giữ đà này, Sprint sẽ **trượt Sprint Goal**, hoặc phải “rush” cuối Sprint gây bug/nợ kỹ thuật.

2) Scrum Master can thiệp thế nào (làm gì – hỗ trợ ai – làm khi nào – đúng tinh thần Scrum)

Can thiệp cho Vấn đề 1 (Daily không bám Sprint Goal)

- **SM làm gì:** mở Daily bằng 1 câu nhắc mục tiêu 24h gắn Sprint Goal: “Hôm nay cần chạy end-to-end đặt lịch + gửi nhắc lịch trên staging”.
- **Hỗ trợ ai:** **PO + Dev** (để điều chỉnh Sprint Backlog theo luồng giá trị).
- **Làm khi nào:** ngay **Daily ngày 6**, và **Backlog refinement** (hoặc mini session 20 phút sau Daily).
- **Đúng tinh thần Scrum:** tập trung vào **Sprint Goal**, tăng minh bạch, tự quản.

Can thiệp cho Vấn đề 2 (không nói trở ngại)

- **SM làm gì:** tạo “psychological safety”: nhắc “nêu blocker là để gỡ, không phải để đổ lỗi”; hỏi gợi mở trong Daily: “Có gì đang chờ người khác/đợi môi trường/đợi review không?”
- **Hỗ trợ ai:** **Dev + QA + PO** (nếu blocker là yêu cầu/priority).
- **Làm khi nào:** ngay **Daily**, sau đó SM làm việc 1-1 nhanh với 2 bạn im lặng để tìm impediment thật.
- **Đúng tinh thần Scrum:** SM là **servant leader**, gỡ impediments, cải tiến liên tục.

Can thiệp cho Vấn đề 3 (Daily quá dài)

- **SM làm gì:** set **timebox 15 phút**, mỗi người 60–90s; dùng **parking lot**: vấn đề cần bàn sâu → họp riêng sau Daily với đúng người.
- **Hỗ trợ ai:** cả team (đặc biệt người hay “đào sâu”).
- **Làm khi nào:** ngay **Daily ngày 6**.
- **Đúng tinh thần Scrum:** giữ đúng sự kiện, tối ưu hiệu quả giao tiếp.

Can thiệp cho Vấn đề 4 (Kanban tắc, WIP quá cao)

- **SM làm gì:** đặt **WIP limit** (ví dụ In Progress $\leq 3-4$); áp dụng “**Stop starting, start finishing**”; tổ chức “swarming” để kéo task tới Done; thêm cột “Code Review/Testing” nếu cần để lộ bottleneck.
- **Hỗ trợ ai:** **Dev + QA** (vì thường nghẽn ở review/test).
- **Làm khi nào:** điều chỉnh **ngay trong ngày 6** + chốt policy trong **Sprint Retrospective**.
- **Đúng tinh thần Scrum:** minh bạch quy trình, tối ưu dòng chảy công việc, tự quản.

Can thiệp cho Vấn đề 5 (Burndown đi ngang)

- **SM làm gì:** kiểm tra “Done thật chưa”: DoD có bị dồn vào cuối không; tách story lớn; ưu tiên các hạng mục tạo **luồng đặt lịch end-to-end**; nếu cần thì làm việc với PO để **re-scope** (giữ Sprint Goal).
- **Hỗ trợ ai:** **PO** (ưu tiên/giảm scope), **QA** (đẩy test sớm), **Dev** (chẻ nhỏ, giảm WIP).
- **Làm khi nào:** sau Daily ngày 6 tổ chức **inspection session 20–30 phút** (không biến Daily thành họp dài).
- **Đúng tinh thần Scrum:** empiricism (minh bạch–kiểm tra–thích nghi).

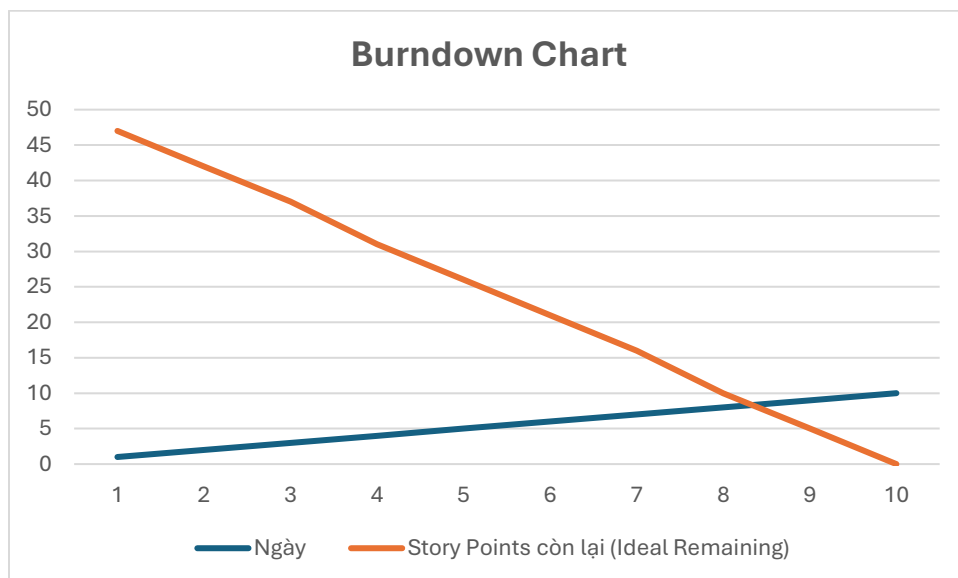
3) Mô phỏng Daily Scrum chuẩn (10 dòng) — ngày thứ 6

1. **SM:** “Mục tiêu hôm nay: hoàn tất luồng *đặt lịch* + trigger *nhắc lịch* trên staging. Mỗi người 3 ý: hôm qua / hôm nay / blocker.”
2. **Dev1:** “Hôm qua em làm UI chọn bác sĩ + chọn giờ. Hôm nay em nối API lấy slot và booking. Blocker: đang chờ API confirm lịch.”
3. **Dev2:** “Hôm qua em làm endpoint tạo lịch hẹn. Hôm nay em hoàn thiện validate + trả lỗi chuẩn. Blocker: thiếu rule ‘trùng slot’ từ PO.”
4. **Dev3:** “Hôm qua em làm service nhắc lịch qua email/push. Hôm nay em tích hợp cron/job và API trigger nhắc. Blocker: env staging chưa có cấu hình gửi mail.”
5. **QA:** “Hôm qua em viết test cases booking + reminder. Hôm nay em test end-to-end trên staging và log bug P0. Blocker: chưa có dữ liệu bác sĩ/slot mẫu ổn định.”
6. **SM:** “Ok, blocker thật sự là *rule trùng slot, mail config staging, data slot mẫu*. Mình không giải quyết chi tiết ở đây.”
7. **SM:** “Sau Daily 10 phút: Dev2 + QA + PO chốt rule ‘trùng slot’. Dev3 + DevOps xử lý mail config.”
8. **SM:** “Dev1 tạm dùng mock slot trong 2 giờ để không đứng việc, nhưng nhớ quay lại tích hợp API confirm.”
9. **SM:** “Nhắc cả nhóm: ưu tiên kéo task về Done, hạn chế mở việc mới khi In Progress đang cao.”
10. **SM:** “Kết thúc daily. Action items mình ghi rồi: rule (PO), config (DevOps), data mẫu (Dev2+QA).”

4) Burndown Chart “lý tưởng” (mô tả bằng chữ)

Giả sử tổng đầu Sprint là **52 points** (vì ngày 1 còn lại 52). Sprint 10 ngày \Rightarrow lý tưởng burn đều khoảng $52/10 \approx 5.2$ points/ngày.

Ngày	Story Points còn lại (Ideal Remaining)
1	47
2	42
3	37
4	31
5	26
6	21
7	16
8	10
9	5
10	0



Đường burndown lý tưởng sẽ giảm đều mỗi ngày (không đi ngang lâu).

Vì sao giúp đạt mục tiêu: giảm đều nghĩa là team đang **hoàn thành “Done”** liên tục, phát

hiện lịch sớm (ngày 3–4 đã thấy), để điều chỉnh WIP/scope/bottleneck để đảm bảo *đặt lịch* + *nhắc lịch* hoàn thành đúng hạn.

Bài 10:

1) Phân tích chuyên sâu nguyên nhân (≥ 5 nguyên nhân rõ ràng)

A. Dấu hiệu từ 3 bộ dữ liệu

- **Daily Scrum:** kéo dài 20 phút; báo cáo kiểu “status report”; **không nhắc Sprint Goal**; không ai nêu **impediments** dù velocity thấp.
- **Kanban:** Day 3 có **To Do 4 / In Progress 6 / Done 0**; nhiều task “kẹt” In Progress > 2 ngày.
- **Burndown:** 32 → 31 → 31 (gần như ngang) ⇒ **không “đóng Done”**.

B. Root causes (nguyên nhân gốc rễ) + hậu quả

1. WIP quá cao (start nhiều – finish ít)

- *Gốc rễ:* không có WIP limit; team kéo cùng lúc nhiều task.
- *Hậu quả:* Task kẹt In Progress, không có Done → burndown đi ngang, rủi ro trượt Sprint Goal.

2. Story/Task quá lớn hoặc cắt chưa đúng

- *Gốc rễ:* T2/T4 (8 SP) có thể quá to; thiếu chia nhỏ thành “thin slice” end-to-end.
- *Hậu quả:* không thể hoàn tất trong 1–2 ngày → không “Done” → flow tắc.

3. Nghẽn ở khâu “hoàn tất” (review/test/integration) nhưng board không thể hiện

- *Gốc rễ:* board chỉ 3 cột nên bottleneck (review/testing) bị giấu; QA tham gia muộn.
- *Hậu quả:* dev “code xong” nhưng không Done; bug/integration dồn cuối Sprint.

4. Daily Scrum sai mục đích (không hướng Sprint Goal, không inspect-adapt)

- *Gốc rễ*: mọi người coi Daily là báo cáo; không dùng dữ liệu Kanban/Burndown để điều chỉnh kế hoạch 24h.
- *Hậu quả*: mỗi ngày trôi qua mà ưu tiên không đổi → vấn đề lặp lại.

5. Impediments bị “ẩn” (không có psychological safety / không biết thế nào là blocker)

- *Gốc rễ*: ngại nói khó khăn; hoặc nghĩ “chờ review/chờ API” không phải blocker.
- *Hậu quả*: Scrum Master không gỡ được trở ngại → velocity thấp kéo dài.

6. Definition of Done mơ hồ / quá nặng cuối chuỗi

- *Gốc rễ*: chưa thống nhất DoD (code + review + test + deploy?) nên “xong” không được tính Done.
- *Hậu quả*: Done = 0; burndown không giảm; chất lượng rơi vào phút chót.

2) Kế hoạch can thiệp trong 24 giờ của Scrum Master (ngày thứ 4)

2.1 Làm gì ngay trong Daily Scrum ngày 4 (giữ đúng 15 phút)

- **Mở đầu 30s**: nhắc Sprint Goal + mục tiêu 24h:
“Hôm nay phải *Done* ít nhất 1–2 hạng mục để khơi lại flow.”
- **Dùng Kanban làm trung tâm**: hỏi theo thẻ đang kẹt, không hỏi lan man.
- **Ép nêu impediments bằng câu hỏi đúng**:
 - “Task nào bị kẹt > 1 ngày? đang chờ ai/cái gì?”
 - “Task nào gần Done nhất để kéo qua Done hôm nay?”
- **Chốt 1 quyết định ngay**: “Stop starting, start finishing” (không kéo task mới nếu chưa kéo được task về Done).

2.2 Làm gì sau Daily (các họp ngắn đúng người, 10–30 phút)

1. **Kanban Rescue (15 phút)**: cả team nhìn board → đặt WIP limit + chọn 2 task “đẩy về Done” trong ngày.
2. **Backlog/Scope sync với PO (15 phút)**:

- Ưu tiên theo Sprint Goal: *tạo lớp* → *mời học viên* → *điểm danh* (thin slice).
- Nếu cần: giảm scope UI đẹp / tối ưu sau; tập trung luồng chạy được.

3. QA + Dev Test Strategy (20 phút):

- Đưa test sớm; tạo checklist DoD; làm test case tối thiểu cho từng task.

4. Gỡ impediment bên ngoài (Scrum Master tự đi xử lý): môi trường test, quyền truy cập, API contract...

2.3 Điều chỉnh Kanban + luật WIP + “khởi lại flow”

- **Bắt buộc đặt WIP limit** cho In Progress (và thêm cột nếu cần).
- **Swarming:** 2–3 người hợp lực kéo 1 thẻ qua Done thay vì mỗi người ôm 1 thẻ.
- **Không mở việc mới khi:**
 - In Progress đã chạm WIP limit, hoặc
 - không có capacity QA/review.
- **Bắt đầu “kéo Done mỗi ngày”:** mục tiêu Day 4 có Done > 0 để phá trạng thái “tắc”.

3) Daily Scrum mẫu (10–12 dòng) — ngày thứ 4 (3 dev + 1 QA)

1. **SM:** “Sprint Goal: tạo lớp + mời học viên + điểm danh. Mục tiêu hôm nay: kéo *ít nhất 1 task* về Done.”
2. **Dev1:** “Hôm qua em làm UI tạo lớp (T1). Hôm nay em hoàn thiện form + gọi API create class. Blocker: chưa có endpoint ổn định.”
3. **Dev2:** “Hôm qua em làm API tạo lớp (T2) nhưng còn validate. Hôm nay em chốt contract + trả lỗi chuẩn + viết swagger. Blocker: còn thiếu rule bắt buộc của lớp (tên, lịch, giới hạn).”
4. **Dev3:** “Hôm qua em bắt đầu điểm danh (T4) nhưng đang dở. Hôm nay em tạm dừng mở mới, chuyển sang hỗ trợ T2 để T1 chạy end-to-end. Blocker: chưa có dữ liệu lớp mẫu để test.”
5. **QA:** “Hôm qua em chưa test được vì chưa có bản deploy. Hôm nay em viết test case tối thiểu cho tạo lớp + chạy Postman. Blocker: thiếu môi trường/staging ổn định.”
6. **SM:** “Mình ghi nhận 3 blocker: rule nghiệp vụ, endpoint ổn định, môi trường test.”

7. **SM:** “Daily không giải quyết chi tiết. Sau họp 15 phút: Dev2 + QA + PO chốt rule và contract.”
 8. **SM:** “Dev1 và Dev2 ưu tiên end-to-end cho T1+T2. Dev3 hỗ trợ data mẫu/seed.”
 9. **SM:** “Nhắc WIP: hôm nay chỉ tập trung T1+T2 để có Done. Không kéo task mới.”
 10. **SM:** “Kết thúc. Action items: PO chốt rule, Dev2 update swagger, QA chuẩn bị test, SM làm việc với DevOps về staging.”
-

4) Thiết kế lại Kanban + Burndown hợp lý

4A) Kanban Board mới + WIP limit + nguyên tắc vận hành

Thiết kế lại board (để lộ bottleneck):

- To Do | In Dev | Review | Testing | Done

WIP limit đề xuất:

Cột	WIP limit
In Dev	3
Review	2
Testing	2

Task nào cần “kéo về lại To Do”?

- Các task đang In Progress > 2 ngày nhưng **chưa rõ next step** / bị chặn nặng.
- Ưu tiên kéo lại những task **không sát Sprint Goal ngay** (ví dụ đang làm UI danh sách học viên T5 trong khi T2/T3 chưa xong).
- Mục tiêu ngày 4: tập trung **T2 (API tạo lớp) + T1 (UI tạo lớp)** để có luồng tạo lớp chạy được.

Nguyên tắc quản lý task mới

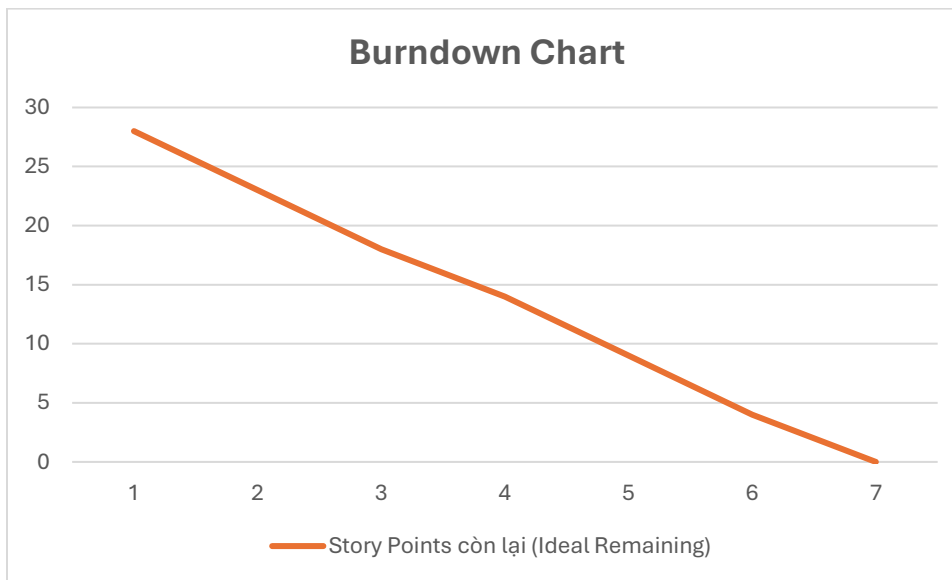
1. **Stop starting, start finishing:** chỉ kéo task mới khi WIP còn chỗ.
2. Task > 1 ngày không nhúc nhích ⇒ đánh cờ đỏ: “Blocked” + ghi rõ lý do.
3. Mỗi ngày phải có **Done > 0** (ít nhất 1 thẻ) để khơi flow.

- Định nghĩa “Done” rõ: code + review + test pass (+ deploy nếu có) mới được qua Done.

4B) Burndown Chart lý tưởng (Sprint 7 ngày, tổng 32 SP)

Burn đều: $32 / 7 \approx 4.6 \text{ SP/ngày}$ \Rightarrow làm tròn theo nhịp thực tế để về 0 ngày 7:

Ngày	Story Points còn lại (Ideal Remaining)
1	28
2	23
3	18
4	14
5	9
6	4
7	0



Vì sao đường lý tưởng giúp Sprint đạt mục tiêu?

- Vì mỗi ngày đều “đóng Done” \rightarrow phát hiện lệch sớm (ngày 2–3 đã thấy), dễ điều chỉnh WIP/ưu tiên/impediments.
- Không dồn review/test cuối Sprint \rightarrow chất lượng ổn, giảm rủi ro trượt Sprint Goal.