

Bài 1:

Hội thoại Sprint Review (Dự án: Ứng dụng quản lý nhóm TeamHub)

PO:

“Chào mọi người. Sprint này nhóm đặt mục tiêu hoàn thành *tính năng tạo nhóm và mời thành viên* để giúp người dùng bắt đầu làm việc trong TeamHub nhanh hơn. Bây giờ mời Dev demo.”

Developer:

“Vâng. Đây là giao diện tạo nhóm mới. Người dùng nhập tên nhóm, mô tả, sau đó bấm **Create**. Nhóm được lưu và hiển thị trong danh sách. Tiếp theo, người dùng có thể nhập email để mời thành viên. Hệ thống sẽ gửi thông báo mời ngay lập tức. Tính năng đã hỗ trợ validate và kiểm tra email trùng.”

Stakeholder:

“Tính năng hoạt động mượt. Tôi muốn bổ sung thêm bước xem trước danh sách email trước khi gửi lời mời, để tránh mời nhầm. Nhóm có thể cân nhắc đưa vào Sprint sau không?”

PO:

“Cảm ơn phản hồi, tôi sẽ ghi nhận và đưa vào Product Backlog để sắp xếp ưu tiên.”

Bài 2:

a. Product Backlog Item (PBI)

User Story – Bộ lọc xem thống kê theo ngày

Role – Goal – Benefit:

- **Role:** Là *trưởng phòng vận hành*,
- **Goal:** tôi muốn xem thống kê hoạt động theo *ngày* hoặc *trong 24 giờ gần nhất*,
- **Benefit:** để tôi có thể kiểm tra chi tiết hoạt động và phát hiện các bất thường nhanh hơn.

Business Requirement (tùy chọn, nếu muốn mô tả thêm):

Hệ thống cần bổ sung bộ lọc theo ngày và bộ lọc theo khoảng thời gian 24 giờ gần nhất trên màn hình thống kê để người dùng xem được dữ liệu chi tiết hơn thay vì chỉ theo tuần.

b. Acceptance Criteria (AC)

Cách viết 1 — Checklist (ngắn gọn, dễ hiểu)

- Có tùy chọn lọc thống kê theo ngày (chọn 1 ngày cụ thể).

- Có tùy chọn “24 giờ gần nhất”.
 - Khi áp dụng bộ lọc, hệ thống hiển thị đúng dữ liệu chi tiết theo khoảng thời gian đã chọn.
 - Nếu không có dữ liệu trong ngày được chọn, hệ thống thông báo “Không có dữ liệu”.
 - Bộ lọc sẽ không ảnh hưởng đến các bộ lọc khác hiện có (ví dụ tuần/tháng).
-

Cách viết 2 — Given – When – Then (chuẩn bài bản Agile)

AC1 – Lọc thống kê theo ngày

- **Given** người dùng đang ở màn hình xem thống kê,
- **When** người dùng chọn một ngày cụ thể trong bộ lọc,
- **Then** hệ thống hiển thị thống kê chi tiết của đúng ngày đó.

AC2 – Lọc thống kê trong 24 giờ gần nhất

- **Given** người dùng mở màn hình thống kê,
- **When** người dùng chọn tùy chọn “24 giờ gần nhất”,
- **Then** hệ thống hiển thị dữ liệu thống kê chính xác trong 24 giờ tính từ thời điểm hiện tại.

AC3 – Không có dữ liệu

- **Given** người dùng chọn một ngày nhưng ngày đó không có hoạt động,
- **When** hệ thống thực hiện truy vấn dữ liệu,
- **Then** hệ thống hiển thị thông báo “Không có dữ liệu”.

Bài 3

a. Timeline Sprint (14 ngày)

Sprint Goal: Hoàn thiện tính năng *Dashboard thống kê nhiệm vụ* và cải thiện hiệu suất load dữ liệu.

Timeline:

- **Ngày 1:** Sprint Planning + chia task.
- **Ngày 2:** Dev bắt đầu tạo API lấy số lượng nhiệm vụ theo trạng thái.
- **Ngày 3:** Hoàn thiện UI chart thống kê.

- **Ngày 4:** Tích hợp API vào UI.
 - **Ngày 5:** Xuất hiện bug load dữ liệu chậm → điều tra nguyên nhân.
 - **Ngày 6:** Tối ưu query & caching API.
 - **Ngày 7:** Hoàn thành bản demo tính năng thống kê Dashboard.
 - **Ngày 8:** Tester phát hiện lỗi hiển thị sai số nhiệm vụ đã hoàn thành.
 - **Ngày 9:** Fix lỗi + viết unit test.
 - **Ngày 10:** Stakeholder yêu cầu thêm bộ lọc theo thời gian (ngày/tuần).
 - **Ngày 11:** PO đánh giá request và đưa vào Product Backlog.
 - **Ngày 12:** Refactor code & dọn dẹp technical debt.
 - **Ngày 13:** Chuẩn bị tài liệu demo Sprint Review.
 - **Ngày 14:** Sprint Review + Sprint Retrospective.
-

b. 5 sự kiện quan trọng trong Sprint

1. Ngày 5 – Phát hiện bug load dữ liệu quá chậm

Mô tả: Dashboard mất hơn 5 giây để tải dữ liệu thống kê.

Ảnh hưởng: Tiêu cực – không thể demo đúng tiến độ, nguy cơ chậm Sprint Goal.

Bài học: Luôn kiểm tra performance ngay từ giai đoạn đầu tích hợp.

2. Ngày 7 – Hoàn thành bản demo Dashboard

Mô tả: Dev hoàn thiện UI + API và chạy thử phiên bản đầy đủ đầu tiên.

Ảnh hưởng: Tích cực – nhóm có thể kiểm thử sớm và nhận phản hồi sớm.

Bài học: Hoàn thành bản demo sớm giúp giảm rủi ro cuối Sprint.

3. Ngày 8 – Tester phát hiện dữ liệu thống kê không khớp

Mô tả: Số lượng nhiệm vụ hiển thị trên biểu đồ không khớp với backend.

Ảnh hưởng: Tiêu cực – phải fix lỗi, mất thêm gần 1 ngày.

Bài học: Cần thống nhất logic thống kê giữa backend và frontend trước khi viết code.

4. Ngày 10 – Stakeholder yêu cầu tính năng mới (bộ lọc theo thời gian)

Mô tả: Stakeholder muốn bộ lọc theo ngày/tuần cho Dashboard.

Ảnh hưởng: Trung tính – yêu cầu tốt, nhưng không kịp đưa vào Sprint này.

Bài học: Luôn bám sát Sprint Goal và tránh mở rộng phạm vi giữa Sprint.

5. Ngày 12 – Refactor & dọn technical debt

Mô tả: Dev cải thiện cấu trúc code, tối ưu API và giảm lỗi.

Ảnh hưởng: Tích cực – giúp Sprint sau phát triển nhanh hơn và giảm bug.

Bài học: Dành thời gian refactor trong Sprint là cần thiết để giữ chất lượng dự án.

Bìa 4:

Kỹ thuật chọn: *Mad – Sad – Glad*

Giải thích (5–7 dòng):

Mad – Sad – Glad phù hợp nhất vì team đang có nhiều cảm xúc tiêu cực (stress, bức xúc, mệt mỏi, mất kết nối). Kỹ thuật này tạo không gian an toàn để mọi người trút cảm xúc thật — điều mà team đang thiếu vì “không ai chủ động chia sẻ vấn đề trong Daily”. Nó giúp Scrum Master hiểu rõ nguyên nhân gốc: điều gì khiến team *Mad* (ức chế), điều gì làm họ *Sad* (buồn, đuối), và điều gì còn *Glad* (điểm sáng hiếm hoi để nuôi lại tinh thần). Những cảm xúc này chính là gốc của việc giao tiếp kém và bug tích tụ.

Các kỹ thuật khác ít phù hợp: *Start–Stop–Continue* và *4Ls* thiên về hành động, không chạm sâu vào cảm xúc; *Sailboat* phù hợp hơn khi team có tinh thần hợp tác và muốn hướng tới mục tiêu chung; *Timeline* lại mang tính phân tích sự kiện, không giúp khai mở cảm xúc vốn đang bị dồn nén của team.

Bài 5:

(A) Mục tiêu của buổi Sprint Review

- Xác nhận rằng tính năng **Tìm kiếm sản phẩm** hoạt động đúng theo yêu cầu của Sprint Goal: hiển thị kết quả nhanh, chính xác và hỗ trợ bộ lọc.
- Thu thập phản hồi từ Stakeholders về tốc độ tìm kiếm, mức độ liên quan của kết quả và UI bộ lọc sản phẩm.
- Kiểm tra mức độ phù hợp với kỳ vọng người dùng và xem có cần cải tiến điều kiện lọc, gợi ý từ khóa, hay tối ưu hiệu suất.

- Quyết định xem tính năng có sẵn sàng đưa vào phiên bản tiếp theo hay cần bổ sung hạng mục vào Product Backlog.
-

✓ (B) Ai sẽ nói gì trong 1 phút mở đầu cuộc họp

Product Owner nói gì?

“Chào mọi người, cảm ơn đã tham gia buổi Sprint Review. Sprint này, mục tiêu chính của team là hoàn thiện tính năng *Tìm kiếm sản phẩm*, một trong những tính năng quan trọng giúp người dùng dễ dàng tìm thấy đúng sản phẩm họ cần. Chúng tôi muốn xác nhận tính năng đã đáp ứng đúng mục tiêu và lắng nghe phản hồi từ các bên liên quan để cải thiện trong Sprint tiếp theo.”

Scrum Master nói gì?

“Trước khi bắt đầu, tôi xin nhắc lại Format buổi Review: nhóm Dev sẽ trình bày những gì đã hoàn thành, Stakeholders có thể đặt câu hỏi hoặc góp ý bất cứ lúc nào. Thời lượng buổi họp là 25 phút và chúng ta sẽ tập trung vào kết quả của Sprint, không bàn về giải pháp kỹ thuật sâu. Nếu không có gì thêm, mời Dev team bắt đầu demo.”

Dev Lead nói gì?

“Cảm ơn mọi người. Hôm nay chúng tôi sẽ demo tính năng *Tìm kiếm sản phẩm* gồm ba phần: nhập từ khóa, hiển thị kết quả và sử dụng bộ lọc nâng cao. Chúng tôi cũng sẽ mô tả một số tình huống lỗi, chẳng hạn khi không tìm được sản phẩm nào. Bây giờ tôi sẽ bắt đầu demo trên môi trường staging.”

✓ (C) Dev sẽ demo tính năng như thế nào (6–8 câu chi tiết)

1. Dev mở trình duyệt và truy cập vào màn hình trang chủ nơi có thanh tìm kiếm ở phần header.
2. Dev nhập từ khóa mẫu, ví dụ “tai nghe bluetooth”, và nhấn Enter để hệ thống thực hiện truy vấn.
3. Trang kết quả hiển thị danh sách 12 sản phẩm phù hợp, kèm theo hình ảnh, giá và đánh giá người dùng.
4. Dev giải thích rằng kết quả được sắp xếp theo mức độ liên quan, sau đó trình bày cách người dùng có thể lọc theo mức giá, thương hiệu hoặc đánh giá sao thông qua sidebar bên trái.

5. Dev chọn bộ lọc “Giá < 1.000.000đ”, kết quả lập tức cập nhật và chỉ còn 5 sản phẩm đáp ứng điều kiện.
6. Dev mô tả tình huống lỗi: nhập từ khóa “abcxyz123” — hệ thống không tìm thấy kết quả và hiển thị thông báo “Không có sản phẩm phù hợp, vui lòng thử lại từ khóa khác”.
7. Dev cũng demo tính năng gợi ý từ khóa: khi xóa và nhập lại “tai”, hệ thống hiện danh sách gợi ý như “tai nghe”, “tai nghe gaming”, “tai nghe bluetooth”.
8. Cuối cùng, Dev khẳng định tốc độ phản hồi hiện tại dưới 300ms cho truy vấn phổ biến, đạt yêu cầu Sprint Goal, và mời Stakeholders đặt câu hỏi hoặc góp ý thêm.

Bài 6:

(A) Tên Retrospective + kỹ thuật sử dụng

Tên:

“Boost the Flow – Tăng tốc nhịp đội”

Kỹ thuật sử dụng (2 kỹ thuật):

1. **Timeline (10–12 phút)** – giúp team nhìn lại các sự kiện đã xảy ra trong Sprint để lộ rõ nguyên nhân spillover và bug muộn.
2. **Start – Stop – Continue (30 phút)** – giúp team đề xuất hành vi cụ thể để thay đổi, cải thiện quy trình và kỷ luật Daily.

✓ (B) Agenda chi tiết cho buổi Retro (90 phút)

1. Check-in (5 phút)

- Scrum Master hỏi: “Hôm nay bạn mang theo năng lượng ở mức 1–5? Vì sao?”
- Mục tiêu: mở không khí, tạo sự an toàn để chia sẻ.

2. Kỹ thuật 1: Timeline – Nhìn lại Sprint (15 phút)

- 5 phút: SM vẽ timeline lên bảng: Ngày 1 → Ngày 14.
- 10 phút:
 - Team dán note mô tả các sự kiện quan trọng: Daily muộn, task chậm, bug báo trễ, các buổi họp có vấn đề...

- Scrum Master tóm tắt lại 3 điểm nổi bật:
 1. Daily thường bắt đầu trễ → thói quen kém.
 2. Bug báo muộn → QA quá tải, thiếu thời gian test sớm.
 3. Spillover 3 task → ước lượng chưa tốt, hoặc phát hiện yêu cầu muộn.
-

3. Kỹ thuật 2: Start – Stop – Continue (30 phút)

- Chia bảng thành 3 cột: START / STOP / CONTINUE
 - Team viết ý kiến lên sticky note và thảo luận:
 - START: test sớm, chấm công Daily đúng giờ, refinement đều đặn
 - STOP: ôm task quá lớn, báo bug sát deadline
 - CONTINUE: pair-testing, support nhau khi gặp việc
-

4. Thảo luận Root Cause (fishbone hoặc 5 Whys) – 20 phút

- Đi sâu vào 2 vấn đề “Daily muộn” và “Bug báo trễ”.
 - Xác định nguyên nhân cụ thể, thực tế, không đổ lỗi cá nhân.
 - Chốt lại các nhóm nguyên nhân chính để chuyển thành Action Item.
-

5. Chốt Action Items + Người chịu trách nhiệm (15 phút)

- Team bỏ phiếu chọn 5 hành động quan trọng nhất.
 - Giao nhiệm vụ rõ ràng: Ai – Khi nào – Đo thế nào.
-

6. Check-out (5 phút)

- Mỗi người nói 1 câu: “Tôi cam kết gì cho Sprint tới?”
 - Kết thúc Retro.
-

☒ (C) 5 Action Items cụ thể (có deadline – người phụ trách – đo lường được)

1. Daily bắt đầu đúng giờ

- **Ai:** Scrum Master
 - **Làm gì:** Thiết lập rule: Daily bắt đầu đúng 9:00, ai đến sau tự update status vào Slack.
 - **Khi nào:** Áp dụng từ Sprint tới.
 - **Đo lường:** 90% buổi Daily bắt đầu đúng giờ trong Sprint tiếp theo.
-

2. Chia nhỏ task để tránh spillover

- **Ai:** Dev Lead
 - **Làm gì:** Review tất cả task > 8h và bắt buộc chia nhỏ trước khi vào Sprint Backlog.
 - **Khi nào:** Trước Sprint Planning tiếp theo.
 - **Đo lường:** Giảm số task spillover từ 3 → 0 hoặc 1 trong Sprint sau.
-

3. Thiết lập QA Early Testing

- **Ai:** QA Lead
 - **Làm gì:** Thêm rule: bất kỳ task nào đạt 60% sẽ được chuyển sang QA để test sớm phần đã xong.
 - **Khi nào:** Từ ngày đầu Sprint kế tiếp.
 - **Đo lường:** 80% bug được phát hiện trước ngày thứ 10 của Sprint.
-

4. Thêm buổi Mini-Refinement 30 phút giữa Sprint

- **Ai:** Product Owner
 - **Làm gì:** Lên lịch họp refinement 30 phút vào ngày thứ 5 mỗi Sprint.
 - **Khi nào:** Bắt đầu từ Sprint tới.
 - **Đo lường:** Giảm số trường hợp hiểu sai yêu cầu → QA báo lại.
-

5. Pair-testing 1 lần/tuần

- **Ai:** 1 Dev + 1 QA luân phiên
- **Làm gì:** Thực hiện pair-testing với sản phẩm đang hoàn thiện.
- **Khi nào:** Mỗi tuần 1 buổi, 45 phút.
- **Đo lường:** Giảm bug nghiêm trọng (Major) xuống ít nhất 30%.

Bài 7:

Format	Thời gian đề xuất	Phù hợp khi team...	Ưu điểm	Nhược điểm	Điểm dễ dẫn dắt (1–5)
Start – Stop – Continue	45–60 phút	Team muốn hành động nhanh, thực tế; Team hơi im lặng nhưng không căng thẳng	Dễ hiểu, trực tiếp, tạo action rõ ràng	Ít đào sâu cảm xúc; dễ tạo danh sách quá dài	5 (dễ nhất)
Sailboat	60–90 phút	Team fatigue, thiếu năng lượng; cần nhìn tổng thể Sprint; muốn khơi gợi nói thật	Hình ảnh trực quan, vui; giúp phân tích mục tiêu, cản trở, rủi ro	Tốn thời gian; dễ lan man nếu SM không kiểm soát	3
Mad – Sad – Glad	45–60 phút	Team có nhiều cảm xúc bị dồn nén; không khí hơi “âm”; cần giải tỏa	Giúp team nói thật, giải phóng cảm xúc; tăng kết nối	Dễ sa vào phàn nàn nếu dẫn dắt kém; ít tạo hành động cụ thể	4
4Ls (Liked, Learned, Lacked, Longed for)	60–75 phút	Team muốn cân bằng giữa cảm xúc – bài học – thiếu sót; phù hợp team trung bình trở lên	Toàn diện: tích cực, học được gì, thiếu gì; dễ tạo insight sâu	Tốn thời gian viết; đôi khi quá “học thuật” → team trẻ dễ chán	3
KALM (Keep, Add, More, Less)	60 phút	Team đã quen Retro; muốn cải tiến quy trình cụ	Kết hợp tốt giữa giữ – cải thiện – giảm; tạo action thực tế	Người mới khó phân biệt Add vs More; cần dẫn dắt rõ	4

		thể, dựa trên dữ liệu			
--	--	-----------------------	--	--	--

Bài 8:

A. Lời mở đầu để lấy lại không khí (trong 1–2 phút)

Scrum Master nói:

“Cả team mình cùng pause 1 phút nhé. Retro là nơi để cải thiện hệ thống làm việc, không phải để đánh giá cá nhân. Tất cả chúng ta đều cùng một mục tiêu: làm Sprint sau tốt hơn. Giờ mình sẽ đổi sang hoạt động viết ý kiến im lặng để mọi người có không gian bình tĩnh lại.”

Tác dụng:

- ✓ Dừng xung đột ngay lập tức
- ✓ Chuyển từ “nói – cãi nhau” sang “viết – suy nghĩ”
- ✓ Reset năng lượng cả phòng

B. Kỹ thuật dùng để tránh đổ lỗi cá nhân

Kỹ thuật: Silent Writing + “Timeline of Events” (Khung thời gian sự kiện)

Cách làm:

1. Silent Writing 5 phút

- Mỗi người ghi ra:
 - Điều gì diễn ra tốt
 - Điều gì chưa tốt
 - Điều gì nên cải thiện
→ Không được ghi tên người, chỉ ghi sự kiện hoặc quy trình.

2. Timeline 10 phút

- Vẽ timeline Sprint lên bảng.
- Mỗi người dán sticky lên đúng thời điểm sự kiện diễn ra.
- Chỉ nói về **quy trình, dòng chảy công việc, dữ liệu**, KHÔNG nói về cá nhân.

👉 Hiệu quả:

- Redirect sự chú ý khỏi “ai sai” → “điều gì xảy ra”

- Phá căng thẳng nhanh vì tất cả nhìn vào sự thật theo dòng thời gian
 - PO cũng không chen ngang vì mọi thứ đã được đặt lên bảng theo thứ tự
-

C. 3 câu nói “thần chú” khi có người đổ lỗi

1. “Mình tập trung vào vấn đề, không phải vào con người nhé.”

Lời nhắc nhẹ nhưng cực hiệu quả.

2. “Điều gì trong quy trình khiến việc này xảy ra?”

Kéo mọi người ra khỏi việc nói về người và chuyển sang nói về hệ thống.

3. “Nếu không nêu tên ai, sự việc sẽ được mô tả như thế nào?”

Forced reframing → buộc mọi người đổi ngôn ngữ sang hướng trung lập.

D. Kịch bản xử lý chi tiết (phút 1 → phút 20)

Phút 1–2: Reset không khí

Scrum Master dừng mọi trao đổi → nói câu mở đầu phần A.

Phút 3–7: Silent Writing

“Giờ mọi người viết ý kiến cá nhân 5 phút nhé, không nói chuyện, không nhắc tên ai.”

Phút 8–18: Timeline Mapping

Scrum Master:

“Mời mọi người dán note lên timeline. Giờ chúng ta chỉ nói về sự kiện và dữ liệu thực tế.”

Nếu Developer A đổ lỗi tiếp:

- Scrum Master dùng “thần chú số 1”.

Nếu Tester B phản ứng lại:

- Dùng “thần chú số 2”.

Nếu PO chen ngang:

- Scrum Master nói:

“Cảm ơn PO, phần này team đang phân tích quy trình. Chúng ta sẽ nghe ưu tiên kinh doanh sau. Giờ team dev-test cần tiếng nói trước.”

Nếu có người dùng điện thoại:

- Scrum Master mỉm cười, nhẹ nhàng:

“Mình cần full attention 15 phút thôi nhé. Mọi người cùng tham gia để Retro có ý nghĩa.”

Phút 18–20: Chốt vấn đề gốc

Scrum Master:

“Vây root cause mình thấy là: thiếu quy trình phối hợp Dev–Test → Bug báo muộn, spillover xảy ra, và xung đột tăng.”

E. Action Items để giải quyết tận gốc vấn đề đổ lỗi

1. Dev Lead – Tạo checklist “Definition of Ready cho task test” – Trước Sprint sau – Đo bằng: 100% task có đầy đủ thông tin trước khi giao Test.

2. Tester Lead – Thiết lập cutoff time báo bug (ví dụ 16h mỗi ngày) – Bắt đầu từ tuần sau – Đo: 90% bug báo trước cutoff time.

3. Scrum Master – Tổ chức mini workshop “Ngôn ngữ phi đổ lỗi” 30 phút – Trước Sprint Planning – Đo: Không còn ghi tên cá nhân trong Retro tiếp theo.

4. Cả team – Áp dụng rule: “No blame, no name, only events” trong mọi Retro – Áp dụng ngay – Đo: Không có xung đột cá nhân trong 2 Sprint liên tiếp.

5. Dev + Test – Tạo channel Slack riêng để sync lỗi hằng ngày – Setup trong hôm nay – Đo: Giảm 50% bug báo muộn cuối Sprint.