

Bài 1:

Sprint Review

PO:

“Chào mọi người, cảm ơn đã tham dự Sprint Review. Mục tiêu của Sprint lần này là hoàn thiện chức năng tạo và quản lý nhiệm vụ cho từng thành viên, giúp trưởng nhóm dễ giao việc và theo dõi tiến độ.”

Dev:

“Trong Sprint này, nhóm đã hoàn thành tính năng *Tạo nhiệm vụ* (*Task Creation*). Em xin demo nhanh:

- Trưởng nhóm bấm ‘Tạo nhiệm vụ’.
 - Nhập tiêu đề, mô tả, deadline, chọn người phụ trách.
 - Nhấn ‘Lưu’, nhiệm vụ lập tức xuất hiện trong bảng Kanban.
- Ngoài ra, chúng em cũng bổ sung thông báo tự động khi nhiệm vụ được giao.”

Stakeholder:

“Tính năng chạy mượt và trực quan đấy. Tuy nhiên tôi đề xuất thêm tùy chọn gắn mức độ ưu tiên và cho phép chỉnh sửa nhiệm vụ sau khi tạo, để tối ưu quy trình giao việc. Có thể đưa vào Sprint sau được không?”

PO:

“Cảm ơn góp ý. Hai yêu cầu này hợp lý, tôi sẽ đưa vào Product Backlog và ưu tiên trong buổi Refinement tuần sau.”

Bài 2:

a. Product Backlog Item (PBI)

Title: Thêm bộ lọc thống kê theo ngày

User Story (Role – Goal – Benefit):

As a trưởng phòng vận hành,

I want to lọc thống kê theo ngày hoặc trong 24 giờ gần nhất,

So that I can xem chi tiết hoạt động của người dùng chính xác hơn.

b. Acceptance Criteria

Dạng checklist

- Hệ thống cho phép chọn **ngày cụ thể** để xem thống kê.
- Hệ thống cho phép chọn **thống kê trong 24 giờ gần nhất**.
- Khi thay đổi bộ lọc, dữ liệu thống kê được **cập nhật đúng theo lựa chọn**.

Bài 3:

a. Timeline Sprint (Ngày 1 → Ngày 14)

Sprint 2 – Thời gian: 14 ngày

- **Ngày 1:** Sprint Planning – xác định mục tiêu và chọn backlog.
- **Ngày 2:** Bắt đầu phát triển tính năng “Tạo nhiệm vụ”.
- **Ngày 3:** Dev hoàn thiện UI cơ bản.
- **Ngày 4:** Tích hợp API tạo nhiệm vụ.
- **Ngày 5:** Kiểm thử nội bộ lần 1.
- **Ngày 6:** Bắt đầu phát triển tính năng “Bộ lọc nhiệm vụ”.
- **Ngày 7:** Gặp bug đồng bộ dữ liệu.
- **Ngày 8:** Họp nội bộ để xử lý bug & điều chỉnh plan.
- **Ngày 9:** Fix xong bug, tiếp tục phát triển bộ lọc.
- **Ngày 10:** QA test lại toàn bộ.
- **Ngày 11:** Stakeholder yêu cầu bổ sung trường “độ ưu tiên”.
- **Ngày 12:** Dev bổ sung UI + logic cho trường ưu tiên.
- **Ngày 13:** Hoàn thiện, chạy regression test.
- **Ngày 14:** Sprint Review + Retrospective.

b. Năm sự kiện quan trọng trong Sprint

1. Ngày 4 – Hoàn thành API tạo nhiệm vụ

Mô tả: Backend tích hợp xong API để tạo nhiệm vụ mới.

Ảnh hưởng: Tích cực – Dev frontend có thể hoàn thiện luồng end-to-end.

Bài học: Hoàn thành sớm các phần phụ thuộc giúp giảm rủi ro dồn việc cuối sprint.

2. Ngày 7 – Xuất hiện bug đồng bộ dữ liệu giữa client & server

Mô tả: Nhiệm vụ mới tạo không hiển thị ngay do cache không cập nhật.

Ảnh hưởng: Tiêu cực – chậm tiến độ 1 ngày.

Bài học: Cần test kỹ các luồng real-time ngay từ đầu thay vì để về sau.

3. Ngày 8 – Họp nội bộ điều chỉnh kế hoạch Sprint

Mô tả: Nhóm họp 30 phút để đánh giá bug lớn và quyết định ưu tiên fix trước.

Ảnh hưởng: Tích cực – giảm stress cho team và plan được kiểm soát lại.

Bài học: Giao tiếp kịp thời giúp team không đi lệch hướng.

4. Ngày 11 – Stakeholder yêu cầu thêm trường độ ưu tiên

Mô tả: Yêu cầu mới xuất hiện khi xem demo bản tạm.

Ảnh hưởng: Vừa tích cực vừa tiêu cực – tăng giá trị sản phẩm nhưng đội tốn thêm effort ngoài plan.

Bài học: Cần quản lý scope cẩn thận và thương lượng rõ trước khi nhận thêm việc.

5. Ngày 13 – Regression test phát hiện UI không đồng nhất

Mô tả: QA phát hiện giao diện “Độ ưu tiên” chưa thống nhất với style của app.

Ảnh hưởng: Tiêu cực nhẹ – mất thêm 2 giờ sửa.

Bài học: Cần checklist UI chuẩn trước khi dev bắt đầu để tránh sửa lại.

Bài 4:

Em chọn kỹ thuật: *Mad – Sad – Glad*

Lý do chọn :Kỹ thuật **Mad – Sad – Glad** phù hợp nhất trong bối cảnh team đang “âm” và stress vì nó tập trung vào **giải tỏa cảm xúc thật** của từng người — điều mà team hiện tại đang né tránh trong Daily. Cách phân loại theo 3 nhóm cảm xúc giúp mọi người nói ra điều khiến họ bức (Mad), buồn (Sad) và vẫn còn tích cực (Glad), từ đó phá băng và mở ra sự chia sẻ chân thật. Khi cảm xúc được khai thông, team mới đủ an toàn tâm lý để bàn sâu về nguyên nhân gốc rễ của stress và bug dồn cuối Sprint. So với các kỹ thuật khác, Start–Stop–Continue và 4Ls hơi “khô”, tập trung vào hành động hơn cảm xúc nên khó khai mở vấn đề thật. Sailboat thì phù hợp cho tầm nhìn dài hạn, không bám sát cảm xúc tiêu cực hiện tại. Timeline tốt cho phân tích lịch sử nhưng không giúp giải tỏa tâm lý đang đè nặng.

Bài 5:

A. Mục tiêu của buổi Sprint Review

- Xác nhận tính năng **Tạo bài tập** hoạt động đúng như yêu cầu trong Product Backlog.
- Thu thập phản hồi từ **giảng viên và quản trị viên** về giao diện, các trường thông tin, và luồng tạo bài tập.
- Kiểm tra khả năng xử lý dữ liệu đầu vào không hợp lệ hoặc thiếu thông tin.
- Ưu tiên cải tiến hoặc bổ sung nếu có, để lên kế hoạch cho Sprint tiếp theo.
- Đảm bảo team và stakeholders hiểu rõ tiến độ, kết quả của Sprint.

B. Ai sẽ nói gì trong 1 phút mở đầu cuộc họp

Product Owner (PO):

“Chào mọi người, trong Sprint này, Sprint Goal của chúng ta là hoàn thiện tính năng **Tạo bài tập mới** cho hệ thống LMS. Tính năng này giúp giảng viên dễ dàng tạo, chỉnh sửa, và gửi bài tập cho sinh viên, đồng thời hỗ trợ quản trị viên kiểm soát toàn bộ danh sách bài tập. Mục tiêu buổi Review hôm nay là demo tính năng và lấy phản hồi để xem có cần bổ sung gì cho Sprint tiếp theo.”

Scrum Master (SM):

“Cảm ơn PO. Nhắc mọi người, buổi Review hôm nay kéo dài khoảng 45 phút. Chúng ta giữ quy tắc: ai muốn đặt câu hỏi hoặc đưa ý kiến, vui lòng ghi ra hoặc giơ tay để giữ trật tự. Dev sẽ demo tính năng trước, sau đó mới thảo luận và feedback.”

Dev Lead:

“Xin chào tất cả, nhóm sẽ demo tính năng **Tạo bài tập** gồm các bước từ mở giao diện, nhập dữ liệu, lưu

bài tập và xử lý lỗi nếu thông tin chưa hợp lệ. Demo sẽ minh họa cả luồng thành công và tình huống lỗi để các bạn có thể hình dung rõ ràng.”

C. Dev sẽ demo tính năng như thế nào

1. **Bước mở giao diện:** Dev click vào menu “Assignments” → chọn “Create New Assignment”. Giao diện tạo bài tập xuất hiện với các trường thông tin bắt buộc: tiêu đề, mô tả, ngày nộp, loại bài tập.
2. **Dữ liệu mẫu:** Dev điền tiêu đề “Bài tập 1 – Toán”, mô tả ngắn, chọn ngày nộp là 05/12/2025, loại bài tập “Essay”.
3. **Luồng chính:** Nhấn nút “Save”. Hệ thống kiểm tra dữ liệu đầu vào. Nếu hợp lệ, bài tập được lưu và hiển thị trong danh sách bài tập của giảng viên.
4. **Hiển thị kết quả:** Danh sách bài tập tự động cập nhật, hiển thị tiêu đề, loại bài tập, ngày nộp, trạng thái “Active”.
5. **Tình huống lỗi:** Dev thử để trống tiêu đề → nhấn “Save”. Hệ thống hiển thị thông báo lỗi: “Tiêu đề bài tập không được để trống”.
6. **Tình huống lỗi khác:** Chọn ngày nộp trước ngày hiện tại → hệ thống cảnh báo: “Ngày nộp phải sau ngày hiện tại”.
7. **Kết quả mong đợi:** Stakeholder thấy rõ luồng tạo bài tập, các kiểm tra lỗi hoạt động, giao diện thân thiện, thông tin hiển thị đầy đủ.
8. **Tương tác cuối demo:** Dev mở tính năng chỉnh sửa bài tập vừa tạo, cho thấy việc update thông tin diễn ra mượt mà, chuẩn bị cho bước phản hồi của PO và stakeholders.

Bài 6:

A. Tên Retrospective + Kỹ thuật sử dụng

Tên buổi: Sprint Retrospective – Sprint 6

Kỹ thuật sử dụng:

- **Mad – Sad – Glad** → để khai thác cảm xúc và giải tỏa tension từ các vấn đề trong Sprint.
- **Start – Stop – Continue** → để tập trung vào hành động cải tiến cụ thể cho Sprint tiếp theo.

B. Agenda chi tiết (90 phút)

Thời gian	Hoạt động	Mục tiêu
0 – 10'	Check-in + Icebreaker	Mọi người chia sẻ 1 cảm xúc hiện tại bằng 1 từ, tạo không khí mở.
10 – 35'	Mad – Sad – Glad	<ul style="list-style-type: none"> - Mỗi thành viên ghi ra điều khiến họ Mad (bực bội), Sad (buồn) và Glad (vui). - Chia sẻ lần lượt, SM ghi lại trên board. - Giúp team giải tỏa cảm xúc, tạo an toàn tâm lý.
35 – 50'	Nhóm và thảo luận root-cause	SM hướng dẫn team phân tích nguyên nhân gốc rễ các vấn đề chính (ví dụ: Daily muộn, spillover, bug cuối Sprint).
50 – 75'	Start – Stop – Continue	<ul style="list-style-type: none"> - Mỗi thành viên đưa ra hành động cần Start / Stop / Continue. - SM tổng hợp và ưu tiên 5–7 hành động khả thi cho Sprint tiếp theo.
75 – 85'	Định nghĩa Action Items	Xác định Ai – Làm gì – Khi nào – Đo lường thế nào cho mỗi Action Item.
85 – 90'	Check-out	Mỗi người chia sẻ 1 câu về cảm giác sau buổi Retro, kết thúc buổi.

C. 5 Action Items cụ thể

Ai	Làm gì	Khi nào	Đo lường
Scrum Master	Gửi nhắc nhở 15 phút trước Daily để team bắt đầu đúng giờ	Từ Sprint tiếp theo	Kiểm tra thời gian bắt đầu Daily mỗi ngày
Dev Lead	Lập checklist task trước Sprint Review để hạn chế spillover	Sprint tiếp theo	Số task spillover ≤ 1
Tester	Thử kiểm tra sớm các tính năng mới, báo bug ít nhất 2 ngày trước khi kết thúc Sprint	Sprint tiếp theo	Số bug báo muộn = 0
Dev Team	Dành 10 phút cuối Daily để mọi người chia sẻ ý kiến cải tiến	Sprint tiếp theo	Daily có ít nhất 2 ý kiến cải tiến được ghi nhận
Scrum Master & PO	Tổ chức mini-retro giữa Sprint (mid-Sprint check) để phát hiện vấn đề sớm	Giữa Sprint	Có ít nhất 1 điều chỉnh kịp thời nếu phát sinh vấn đề

Bài 7:

Format	Thời gian đề xuất	Phù hợp khi team...	Ưu điểm	Nhược điểm	Điểm số dễ dẫn dắt (1-5)
Start – Stop – Continue	60–90 phút	Muốn xác định hành động cải tiến rõ ràng, team mới hoặc có vấn đề cụ thể cần giải quyết	Dễ hiểu, tập trung vào hành động, giúp Sprint tiếp theo cải tiến cụ thể	Có thể khô cứng, ít khai thác cảm xúc	4
Sailboat	60–90 phút	Team muốn nhìn nhận tổng quan Sprint, rủi ro, cơ hội và mục tiêu dài hạn	Giúp team định hướng, nhận diện rủi ro & động lực, trực quan	Tốn thời gian, ít tập trung vào hành động chi tiết	3
Mad – Sad – Glad	45–60 phút	Team đang stress, cần giải tỏa cảm xúc hoặc có không khí căng thẳng	Khai thác cảm xúc thật, phá băng, tạo an toàn tâm lý	Không tập trung nhiều vào hành động cụ thể, cần dẫn dắt khéo	5
4L (Liked, Learned, Lacked, Longed for)	60–90 phút	Team muốn phản ánh sâu hơn về kinh nghiệm, học hỏi và mong muốn	Khai thác học hỏi, ý kiến sâu sắc, giúp team trưởng thành	Có thể dài dòng, khó tổng hợp thành hành động ngay	4
KALM (Keep, Add, More, Less)	60 phút	Team muốn cân bằng giữa việc duy trì & cải tiến	Giúp team cân nhắc giữ gì, thay đổi gì, dễ trực quan	Chưa tập trung vào cảm xúc, cần giải thích kỹ	3

Bài 8:

A. Lấy lại không khí đầu buổi

Thời gian: 0 – 5 phút

- **Mở đầu:**

“Chào mọi người, hôm nay chúng ta sẽ cùng nhìn lại Sprint vừa rồi để học hỏi và cải thiện, không phải để đổ lỗi hay tranh cãi. Mục tiêu của chúng ta là **cùng nhau tìm giải pháp giúp Sprint tiếp theo hiệu quả hơn**. Ai cũng có ý kiến quan trọng, vì vậy hãy chia sẻ với tinh thần xây dựng.”

- **Giao luật ngắn:**

1. Không đổ lỗi cá nhân.

2. Ai muốn nói, giơ tay hoặc ghi ra giấy.
3. Giữ không khí tôn trọng, không ngắt lời.

B. Kỹ thuật dùng để tránh đổ lỗi cá nhân

Kỹ thuật: Silent Writing + Mad – Sad – Glad

Cách thực hiện:

1. Mỗi người viết riêng trên giấy / sticky note các điểm khiến họ **Mad** (bực mình), **Sad** (buồn) và **Glad** (vui) về Sprint vừa qua.
2. Sau 5–10 phút, từng sticky note được dán lên board mà **không ghi tên** người viết.
3. Nhóm thảo luận dựa trên vấn đề/hiện tượng, **không nhắm vào cá nhân**.

Lợi ích:

- Giải tỏa cảm xúc cá nhân.
- Chuyển sự đổ lỗi thành **vấn đề chung cần cải tiến**.

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

1. “Hãy tập trung vào **vấn đề, không phải con người**.”
2. “Chúng ta cùng chịu trách nhiệm về Sprint, ai cũng có cơ hội cải thiện.”
3. “Mọi ý kiến đều quan trọng, nhưng chúng ta hãy nói về **giải pháp** trước khi tìm lỗi ai.”

D. Action Item để giải quyết tận gốc vấn đề đổ lỗi

Ai	Làm gì	Khi nào	Đo lường
Scrum Master & Dev Lead	Thiết lập quy tắc code review & QA checklist , bao gồm deadline rõ ràng cho việc test & code check	Sprint tiếp theo	Số bug muộn ≤ 1; Daily bắt đầu đúng giờ
Dev Team & Tester	Áp dụng pairing / buddy check : Dev và Tester review cùng nhau ít nhất 1 task mỗi Sprint	Sprint tiếp theo	Bug giảm, không còn tình trạng “code bẩn” hay báo bug muộn
Scrum Master	Giám sát Daily Scrum đúng giờ , nhắc nhở team tập trung, hạn chế điện thoại	Sprint tiếp theo	Daily bắt đầu đúng giờ ≥ 90% số ngày
PO & Team	Cam kết tôn trọng quy trình Retro : không chen ngang, giữ focus	Sprint tiếp theo	Retro hoàn thành đúng agenda, feedback chất lượng