

Quản lý dự án Phát triển Phần mềm - SE358.Q11

Đề tài 2: Ứng dụng quản lý lớp học trực tuyến cho trung tâm ngoại ngữ

Hồ sơ giai đoạn C

02.12.2025

23521224 Trương Hoàng Phúc

23521736 Bùi Văn Tùng

23520657 Vũ Quốc Huy

23520466 Tạ Hoàng Hiệp

23520682 Đỗ Đình Khang

Contents

1. Sprint Backlog (Bùi Văn Tùng)	1
2. Sprint Backlog	1
3. Phân vai Scrum	1
4. Sản phẩm chủ sở hữu (Product Owner)	1
5. Chủ nhân xì trum (Scrum Master)	1
6. Biệt đội phát triển (Development Team)	1
7. Sprint 2: Quản lý lớp học và lịch học	2
8. Sprint Goal	2
9. User Stories	2
10. User Story 1: Giảng viên tạo lớp mới	2
11. User Story 2: Giảng viên đặt lịch học	2
12. User Story 3: Học viên xem danh sách lớp	2
13. User Story 4: Giảng viên phân công giảng viên	3
14. User Story 5: API validation cho lịch học	3
15. User Story 6: Conflict detection cho lịch giảng viên	3
16. User Story 7: Integration class & user profile	3
17. Velocity Sprint 2	4
18. Blockers và Action Items	4
19. Blocker 1: Conflict detection policy chưa rõ	4
20. Blocker 2: API contract chưa thống nhất	4
21. Blocker 3: Authorization chưa rõ	5
22. Blocker 4: Thiếu môi trường staging	5

23. Summary Action Items Sprint 2	5
24. Sprint 3: Quản lý bài tập và chấm điểm	6
25. Sprint Goal	6
26. User Stories	6
27. User Story 1: Giảng viên tạo bài tập	6
28. User Story 2: Giảng viên set deadline	6
29. User Story 3: Học viên xem danh sách bài tập	6
30. User Story 4: Học viên nộp bài	7
31. User Story 5: Giảng viên xem bài nộp	7
32. User Story 6: Giảng viên chấm điểm và comment	7
33. User Story 7: Học viên nhận thông báo điểm	7
34. User Story 8: API validation file upload	8
35. Velocity Sprint 3	8
36. Blockers và Action Items	8
37. Blocker 1: Firebase notification chưa setup	8
38. Blocker 2: API response format chưa thống nhất	8
39. Blocker 3: Naming convention chưa đồng bộ	9
40. Blocker 4: Team member lịch học bận	9
41. Summary Action Items Sprint 3	9
42. Tóm tắt Metrics	10
43. Velocity Trends	10
44. Key Lessons	10
45. Recommendations	10

46. Biên bản Daily Scum 1 (Tạ Hoàng Hiệp)	11
47. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 2)	11
48. Làm gì hôm qua?	11
49. Bữa này gì làm?	11
50. Mặc gì vương cái?	12
51. Tắt bản biên tóm	12
52. Biên bản Daily Scum 2 (Tạ Hoàng Hiệp)	12
53. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 3)	12
54. Làm gì hôm qua?	12
55. Bữa này gì làm?	13
56. Mặc gì vương cái?	13
57. Tắt bản biên tóm	13
58. Sprint Retrospective (Trương Hoàng Phúc)	14
59. Prompt Cho AI	14
60. Những việc làm tốt	14
61. Những việc chưa tốt	14
62. Đánh giá đóng góp từng thành viên	14
63. Phúc (Product Owner)	14
64. Khang (Scrum Master)	15
65. Hiệp (Scrum Master)	15
66. Huy (Developer)	15
67. Tùng (Developer)	15
68. Biện pháp cải tiến	15

69. So sánh Agile vs. Waterfall (Vũ Quốc Huy)	16
70. Phân tích Agile vs Waterfall lĩnh vực Rủi ro	16
71. Em xin lỗi vì đã hỏi cái máy ạ 🙄	16
72. Nếu thực hiện Waterfall thì sao?	16
73. Nếu thực hiện Agile thì sao?	16
74. Bảng so sánh Ưu và Nhược điểm	17
75. Ví dụ minh họa (Case Study)	18
76. Kịch bản 1: Áp dụng Waterfall (Kịch bản thất bại điển hình)	18
77. Kịch bản 2: Áp dụng Agile (Kịch bản thành công)	19
78. Nhận xét mô hình hợp lý nhất	19
79. Lý do:	19
80. Báo cáo cuối kì (Đỗ Đình Khang)	20
81. Giới thiệu dự án	1
81.1. Bối cảnh dự án	1
81.2. Mục tiêu dự án	1
81.3. Phạm vi dự án	1
81.3.1. Các chức năng chính	1
81.3.2. Ranh giới dự án	2
81.4. Stakeholders chính	2
81.5. Lợi ích kỳ vọng	3
82. Tóm tắt tài liệu quản lý	3
82.1. Project Charter (Hiển chương dự án)	3
82.1.1. Thông tin cơ bản	4
82.1.2. Mục đích dự án	4
82.1.3. Sản phẩm bàn giao chính	4
82.1.4. Các mốc quan trọng	4
82.1.5. Ngân sách	4
82.1.6. Quyền hạn Project Manager	5
82.1.7. Rủi ro cấp cao	5
82.2. Work Breakdown Structure (WBS)	5
82.2.1. Giai đoạn 1: Phân tích yêu cầu	5
82.2.2. Giai đoạn 2: Thiết kế hệ thống	5

82.2.3. Giai đoạn 3: Phát triển và Kiểm thử	6
82.2.4. Giai đoạn 4: Viết báo cáo và Thuyết trình	6
82.2.5. Quản lý rủi ro trong WBS	7
82.3. Quản lý Stakeholder	7
82.3.1. Phân loại Stakeholder	7
82.3.2. Chiến lược giao tiếp	8
82.4. Quản lý Chi phí	8
82.4.1. Ước lượng chi phí	8
82.4.2. Baseline chi phí	10
82.4.3. Kiểm soát chi phí	10
83. Phân tích so sánh Agile vs Waterfall	10
83.1. Tổng quan về hai phương pháp	10
83.1.1. Waterfall (Thác nước)	10
83.1.2. Agile (Linh hoạt)	11
83.2. So sánh chi tiết	11
83.3. Áp dụng cho dự án quản lý lớp học	11
83.3.1. Phân tích bối cảnh dự án	11
83.3.2. Đề xuất: Hybrid Approach (Kết hợp)	12
83.3.3. Lợi ích của Hybrid Approach	12
84. Kết quả mô phỏng Sprint	12
84.1. Tổng quan Sprint Planning	12
84.2. Sprint 3: Quản lý bài tập và chấm điểm	13
84.2.1. Sprint Goal	13
84.2.2. Sprint Backlog	13
84.2.3. Sprint Planning Meeting (Ngày 1 Sprint)	13
84.2.4. Daily Standup (Ngày 5 Sprint)	14
84.2.5. Sprint Review (Ngày 14 Sprint - cuối Sprint)	14
84.2.6. Sprint Retrospective (Ngày 14 Sprint - sau Review)	15
84.3. Metrics và Hiệu suất Sprint	15
84.3.1. Burndown Chart	15
84.3.2. Velocity Tracking	15
84.4. Bài học từ Sprint Simulation	16
84.4.1. Về Planning	16
84.4.2. Về Execution	16
84.4.3. Về Collaboration	16
85. Bài học	16
85.1. Về Quản lý Dự án	16
85.1.1. Lập kế hoạch chi tiết từ đầu	16
85.1.2. Quản lý Stakeholder hiệu quả	17
85.1.3. Kiểm soát Scope Creep	17

85.2. Về Phát triển Phần mềm	17
85.2.1. Thiết kế Database từ đầu	17
85.2.2. Code Review và Testing	17
85.2.3. Documentation	18
85.3. Về Teamwork	18
85.3.1. Giao tiếp trong Team	18
85.3.2. Phân công công việc	18
85.3.3. Continuous Improvement	18
85.4. Về Công nghệ	19
85.4.1. Lựa chọn Tech Stack	19
85.4.2. Cloud và Deployment	19
85.5. Khuyến nghị cho Dự án Tương lai	19
85.6. Kết luận	19
 86. Tài liệu tham khảo	 21
 87. Retrospective C	 22
87.1. Đánh giá trải nghiệm Agile	22
87.2. Những khác biệt so với Waterfall	22
87.3. Những việc làm tốt	22
87.4. Những việc làm chưa tốt	22
87.5. Đánh giá đóng góp từng thành viên	22
87.5.1. Hiệp	22
87.5.2. Khang	23
87.5.3. Huy	23
87.5.4. Phúc	23
87.5.5. Tùng	23
87.6. Bài học rút ra	23
87.7. Đề xuất cải thiện	23

1. Sprint Backlog (Bùi Văn Tùng)

2. Sprint Backlog

3. Phân vai Scrum

4. Sản phẩm chủ sở hữu (Product Owner)

Trương Hoàng Phúc (23521224)

- Xác nhận requirements và acceptance criteria cho từng user story
- Chuẩn bị UAT checklist và test cases
- Duyệt demo sản phẩm cuối mỗi Sprint
- Xác định policy khi gặp xung đột yêu cầu

5. Chủ nhân xì trum (Scrum Master)

Đỗ Đình Khang (23520682)

- Tổ chức và facilitate các buổi Daily Scrum, Sprint Planning, Sprint Review
- Kiểm tra code quality thông qua code review
- Đảm bảo Definition of Done được tuân thủ
- Theo dõi velocity của team qua các Sprint

Tạ Hoàng Hiệp (23520466)

- Tổ chức các buổi Daily Scrum và ghi nhận biên bản
- Theo dõi tiến độ integration giữa các module
- Phối hợp với DevOps cho infrastructure setup
- Track blockers liên quan đến infrastructure

6. Biệt đội phát triển (Development Team)

Vũ Quốc Huy (23520657) - Backend Developer

- Phát triển API endpoints cho các tính năng chính
- Xử lý business logic: conflict detection, chấm điểm, validation
- Viết unit test, đảm bảo test coverage $\geq 70\%$
- Deploy code lên staging environment

Bùi Văn Tùng (23521736) - Frontend Developer

- Thiết kế UI/UX flow, tạo wireframe/mockup
- Tích hợp frontend với API backend
- Triển khai validation frontend
- Test upload file, fix UI bugs

7. Sprint 2: Quản lý lớp học và lịch học

8. Sprint Goal

Phát triển chức năng quản lý lớp học, đặt lịch học, phân công giảng viên và xử lý conflict detection.

9. User Stories

10. User Story 1: Giảng viên tạo lớp mới

Người làm: Huy + Tùng | Story Points: 5 | Status: ✓ Hoàn thành

- Huy: Phát triển API POST /classes với field tên lớp, level, sức chứa, giảng viên
- Tùng: Thiết kế UI form tạo lớp, thêm validation frontend
- Khang: Code review, verify test coverage

11. User Story 2: Giảng viên đặt lịch học

Người làm: Huy + Tùng | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: API PUT /classes/{classId}/schedule với ngày, giờ bắt đầu, kết thúc
- Tùng: UI calendar picker, time picker
- Khang: Verify validation edge cases

12. User Story 3: Học viên xem danh sách lớp

Người làm: Huy + Tùng | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: API GET /classes với filter level, teacher name, pagination
- Tùng: Màn hình danh sách lớp dạng card/list, filter UI
- Hiệp: Xác nhận integration test

13. User Story 4: Giảng viên phân công giảng viên

Người làm: Huy + Tùng | Story Points: 5 | Status: ✓ Hoàn thành

- Huy: API PUT /classes/{classId}/assign-teacher, multiple teachers support
- Tùng: UI form dropdown chọn giảng viên, role selector
- Khang: Code review, authorization check

14. User Story 5: API validation cho lịch học

Người làm: Huy | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: Validation giờ kết thúc > giờ bắt đầu, không quá khứ, duplicate check
- Huy: Unit test cover happy path, invalid time range, past schedule, edge cases
- Khang: Merge code, verify test coverage $\geq 70\%$

15. User Story 6: Conflict detection cho lịch giảng viên

Người làm: Huy + Tùng | Story Points: 8 | Status: ⚠ Đang làm

- Huy: Check xung đột lịch khi gán giảng viên, return conflict list
- Tùng: Display warning dialog, option cancel/force assign/reschedule
- Blocker: Phúc chưa xác định policy conflict-resolution

16. User Story 7: Integration class & user profile

Người làm: Hiệp + Huy + Tùng | Story Points: 5 | Status: ○ Chưa làm

- Hiệp: Chuẩn bị integration test plan giữa class + user profile module
- Test: create class → teacher profile update, remove teacher → class unassigned
- Huy + Tùng: Fix bug nếu phát hiện inconsistency

17. Velocity Sprint 2

Committed: 32 points

Completed: 19 points (Stories 1-5)

In Progress: 8 points (Story 6)

Not Started: 5 points (Story 7)

18. Blockers và Action Items

19. Blocker 1: Conflict detection policy chưa rõ

Mô tả: Phúc chưa xác định policy xử lý conflict lịch giảng viên.

Cần xác định:

- Priority rule khi có conflict
- Allow reschedule hay không
- Cách thông báo cho học viên
- Force assign có được phép không

Owner: Phúc (PO)

Timeline: 16/11/2025 EOD

Status: Open

20. Blocker 2: API contract chưa thống nhất

Mô tả: FE & BE chưa align API response format, error response structure, naming convention (snake_case vs camelCase).

Owner: Khang & Hiệp (SMs)

Action:

- Tạo API contract document (Swagger/OpenAPI)
- Standard success response: { status: "success", data: {...}, timestamp: ... }
- Standard error response: { status: "error", code: "...", message: "..." }
- Agree naming convention: camelCase (recommended) hoặc snake_case

Timeline: 17/11/2025

Status: Open

21. Blocker 3: Authorization chưa rõ

Mô tả: Ai có quyền gán giảng viên? Admin only, school manager, hay head teacher?

Owner: Phúc (PO) + Khang (SM)

Action: Phúc xác định policy, Khang implement RBAC vào API middleware

Timeline: 16/11/2025

Status: Open

22. Blocker 4: Thiếu môi trường staging

Mô tả: Team test trên localhost, không có staging server cho integration testing.

Owner: Hiệp (SM) + DevOps

Action:

- Setup staging database (clone schema, sample data)
- Deploy backend code lên staging API server
- Deploy frontend code, point tới staging API
- Setup HTTPS, logs, error tracking

Checklist:

- Database setup: ☐
- API server deploy: ☐
- Frontend host: ☐
- SSL certificate: ☐
- Smoke test: ☐

Timeline: 18/11/2025

Status: Open

23. Summary Action Items Sprint 2

1. Phúc: Xác định conflict-resolution policy (16/11/2025 EOD)
2. Khang & Hiệp: Tạo API contract document (17/11/2025)
3. Khang: Confirm authorization rules với Phúc (16/11/2025)
4. Hiệp: Setup staging environment (18/11/2025)
5. Huy & Tùng: Finalize conflict detection + Story 7 integration (18/11/2025)

24. Sprint 3: Quản lý bài tập và chấm điểm

25. Sprint Goal

Phát triển chức năng quản lý bài tập, nộp bài, chấm điểm và notification khi có điểm mới.

26. User Stories

27. User Story 1: Giảng viên tạo bài tập

Người làm: Huy + Tùng | Story Points: 5 | Status: ✓ Hoàn thành

- Huy: API POST /classes/{classId}/assignments, field title, description, assignment_file, due_date
- Tùng: UI form tạo bài tập, rich text editor, file uploader, datetime picker
- Khang: Code review, file upload security

28. User Story 2: Giảng viên set deadline

Người làm: Huy + Tùng | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: API PUT /assignments/{assignmentId}/deadline, validation deadline không quá khứ
- Tùng: UI dialog set deadline, datetime picker, warning nếu < 24 hours
- Khang: Verify timezone handling

29. User Story 3: Học viên xem danh sách bài tập

Người làm: Huy + Tùng | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: API GET /student/assignments/{classId}, return status, grade, comment
- Tùng: Màn hình danh sách bài tập, status badge (overdue/due soon/submitted), click xem detail
- Hiệp: Test API response format

30. User Story 4: Học viên nộp bài

Người làm: Huy + Tùng | Story Points: 8 | Status: ✓ Hoàn thành

- Huy: API POST /assignments/{assignmentId}/submit, file upload ≤ 10MB, whitelist types (PDF, DOC, DOCX, XLS, XLSX, ZIP, JPG, PNG, GIF)
- Huy: Magic bytes validation, antivirus scan, filename sanitization
- Tùng: UI upload form, drag-drop zone, progress bar, retry logic
- Khang: Verify security, test large files

31. User Story 5: Giảng viên xem bài nộp

Người làm: Huy + Tùng | Story Points: 5 | Status: ✓ Hoàn thành

- Huy: API GET /assignments/{assignmentId}/submissions, filter/sort by student, status
- Tùng: Màn hình danh sách bài nộp table/list, download file, button “Grade”
- Hiệp: Test filtering/sorting logic

32. User Story 6: Giảng viên chấm điểm và comment

Người làm: Huy + Tùng | Story Points: 8 | Status: ⚠ Đang làm

- Huy: API PUT /submissions/{submissionId}/grade, score 0-100, teacher_comment max 1000 chars
- Huy: Publish event “GradePublished” để trigger notification
- Tùng: UI form chấm điểm modal, input score, textarea comment
- Blocker: API chưa finalize, response format chưa match API contract

33. User Story 7: Học viên nhận thông báo điểm

Người làm: Huy + Hiệp + Tùng | Story Points: 5 | Status: ○ Chưa làm

- Huy: Subscribe event “GradePublished”, gọi Firebase FCM gửi notification
- Huy: Notification content “Giáo viên đã chấm [assignment]. Điểm: [score]/100”
- Hiệp: Setup Firebase project, generate FCM server key, config files
- Tùng: Implement FCM client SDK, register device token, handle push notification
- Blocker: Firebase chưa setup

34. User Story 8: API validation file upload

Người làm: Huy | Story Points: 3 | Status: ✓ Hoàn thành

- Huy: File size \leq 10MB (error 413), file type whitelist, magic bytes check, antivirus scan
- Huy: Unit test valid upload, oversized file, invalid type, malicious filename, antivirus fail
- Khang: Merge code, verify coverage \geq 70%

35. Velocity Sprint 3

Committed: 40 points

Completed: 32 points (Stories 1-5, 8)

In Progress: 8 points (Story 6, 70% done)

Not Started: 5 points (Story 7)

36. Blockers và Action Items

37. Blocker 1: Firebase notification chưa setup

Mô tả: DevOps chưa hoàn tất Firebase configuration, ngăn chặn Story 7.

Owner: Hiệp (SM) + DevOps

Action:

- Create Firebase project
- Enable Firebase Cloud Messaging
- Generate FCM server key
- Create config files (google-services.json, GoogleService-Info.plist)
- Setup Firebase Authentication
- Test FCM send test message

Deliverable: Firebase credentials document + config files

Timeline: 17/11/2025 EOD

Status: Open

38. Blocker 2: API response format chưa thống nhất

Mô tả: Một số endpoint return inconsistent response format, error response khác nhau giữa endpoints.

Owner: Khang (SM) + Huy (Backend)

Action:

- Enforce API contract từ Sprint 2
- Standard error response: { status: "error", code: "...", message: "...", details: {...} }
- Huy refactor tất cả endpoints same error format
- Create error codes enum

Timeline: 17/11/2025

Status: Open

39. Blocker 3: Naming convention chưa đồng bộ

Mô tả: Backend snake_case, Frontend camelCase, gây convert confusion.

Owner: Khang (SM) + Huy + Tùng

Action:

- Option 1: Backend return camelCase (recommended)
- Option 2: Frontend transformer snake_case → camelCase (axios interceptor)
- Khang facilitate discussion, decide by 17/11/2025

Timeline: 17/11/2025

Status: Open

40. Blocker 4: Team member lịch học bận

Mô tả: Huy & Tùng có lịch học buổi chiều, giảm productivity.

Owner: Khang & Hiệp (SMs) + Team

Action:

- Shift Daily Scrum sang sáng hoặc tối
- Reduce committed points: plan 30-35 thay vì 40
- Prioritize high-value features
- Pair programming support

Timeline: 17/11/2025

Status: Open

41. Summary Action Items Sprint 3

1. Hiệp: Follow-up Firebase setup (17/11/2025 EOD)
2. Khang: Enforce API response format (17/11/2025)
3. Khang: Facilitate naming convention decision (17/11/2025)

4. Huy: Finalize chấm điểm API (18/11/2025)
5. Huy: Refactor API responses nếu cần (18/11/2025)
6. Tùng: Implement notification UI (18/11/2025)
7. Khang & Hiệp: Adjust Daily Scrum time, re-plan capacity (17/11/2025)
8. Team: Ghi nhận feedback bulk grading → backlog Sprint 4

42. Tóm tắt Metrics

43. Velocity Trends

Sprint 2: 19/32 points completed (60%)
Sprint 3: 32/40 points estimated (80%)
Average velocity: 26 points/sprint
Recommended Sprint 4: 30 points

44. Key Lessons

- Policy/requirements từ PO cần clear trước Sprint: conflict detection policy chưa rõ gây delay
- API contract document critical: FE & BE work in silos nếu không có shared contract
- Infrastructure setup trước sprint: staging environment, Firebase setup không nên delay code
- Communication ngoài Daily Scrum: team cần Slack hoặc async channels
- Respect team capacity: over-commit dẫn đến incomplete sprints

45. Recommendations

- Confirm blockers removed trước Sprint Planning
- Create API contract FIRST, assign tasks sau
- Setup full infrastructure trước development sprints
- Pair programming cho complex features
- Action items từ Retrospective phải có owner + timeline

46. Biên bản Daily Scum 1 (Tạ Hoàng Hiệp)

47. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 2)

Thời gian: 15/11/2025

Thành phần tham gia / Vai trò:

- **Product Owner (PO):** Phúc
- **Scrum Master (SM):** Khang, Hiệp
- **Development Team:** Huy, Tùng

48. Làm gì hôm qua?

- **Phúc (PO):** Xác nhận requirement cho tính năng tạo lớp; ưu tiên các field cần thiết (tên lớp, level, capacity, giảng viên phụ trách).
- **Khang (SM):** Hỗ trợ team rà soát acceptance criteria cho user stories liên quan tới lịch học.
- **Hiệp (SM):** Theo dõi tiến độ integration giữa module class management và user profile.
- **Huy (Dev):** Hoàn thiện API tạo lớp (CRUD) và endpoint phân công giảng viên.
- **Tùng (Dev):** Thiết kế UI flow tạo lớp và form phân công giảng viên; tích hợp validation frontend.

49. Bữa này gì làm?

- **Phúc (PO):** Review demo API tạo lớp và approve acceptance tests; xác nhận policy phân công giảng viên.
- **Khang (SM):** Kiểm tra test cases, đảm bảo unit tests cover core flows; remove blockers nếu có.
- **Hiệp (SM):** Chuẩn bị checklist integration test giữa class management và scheduling service.
- **Huy (Dev):** Bổ sung tính năng update lịch học (time slot) cho lớp, handle conflict detection.
- **Tùng (Dev):** Hoàn thiện UI hiển thị lịch học và chức năng assign giảng viên; fix validation edge cases.

50. Mắc gì vươngng cái?

- **Conflict detection cho lịch học:** Cần chính sách ưu tiên khi giảng viên có xung đột lịch (PO cần đưa ra rule rõ ràng).
 - **API contract:** Một vài endpoint chưa thống nhất response format — cần đồng bộ giữa FE & BE.
 - **Quyền truy cập:** Cần xác định rõ quyền phân công giảng viên (chỉ admin hay giảng viên lead cũng có thể đề xuất?).
 - **Test environment:** Thiếu môi trường staging cho chạy integration tests (SMs action).
-

51. Tắt bản biên tóm

Action items:

1. **Phúc (PO):** Xác định policy conflict-resolution cho lịch giảng viên trong ngày.
2. **Khang / Hiệp (SMs):** Đồng bộ API contract với devs, phối hợp tạo môi trường staging.
3. **Huy / Tùng (Dev):** Hoàn thiện endpoint update lịch và UI assign giảng viên; viết unit & integration tests.

52. Biên bản Daily Scum 2 (Tạ Hoàng Hiệp)

53. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 3)

Thời gian: 17/11/2025 **Thành phần tham gia / Vai trò:**

- **Product Owner (PO):** Phúc
- **Scrum Master (SM):** Khang, Hiệp
- **Development Team:** Huy, Tùng

54. Làm gì hôm qua?

- **Phúc (PO):** Duyệt acceptance criteria cho user story quản lý bài tập; xác nhận giới hạn upload **10MB**.
- **Khang & Hiệp (SM):** Gỡ blockers, theo dõi tiến độ, điều phối code review.

- **Huy (Dev):** Hoàn tất API chấm điểm (draft), thêm validation upload, viết unit test.
- **Tùng (Dev):** Hoàn thành API xem bài nộp, tích hợp frontend, test upload.

55. Bữa này gì làm?

- **Phúc:** Chuẩn bị test case & UAT checklist.
- **Khang:** Kiểm tra CI pipeline, đảm bảo code review.
- **Hiệp:** Theo dõi DevOps để cấu hình Firebase notification.
- **Huy:** Hoàn thiện chấm điểm (bulk grading), unit test, deploy staging.
- **Tùng:** Triển khai notification UI, test upload nhiều định dạng.

56. Mặc gì vương cái?

- **Khang / Hiệp (SMs):** Cần thống nhất naming convention cho cả FE & BE.
 - **Huy / Tùng (Dev):** API response chưa đồng nhất giữa các module.
 - **Phúc (PO):** Cần cập nhật lại timeline Sprint do một số thành viên bận lịch học.
-

57. Tắt bản biên tóm

Action items:

- **Hiệp:** Hoàn tất follow-up Firebase trong ngày.
- **Khang:** Xác nhận CI/CD & quyền truy cập staging.
- **Phúc:** Chuẩn bị UAT checklist.
- **Huy & Tùng:** Hoàn tất feature chấm điểm + notification trên staging hôm nay.
- **Team:** Ghi nhận feedback bulk grading cho Sprint 4.

58. Sprint Retrospective (Trương Hoàng Phúc)

59. Prompt Cho AI

> read the typ file in this workspace, then fill this typ for me > read other file too, not just retrospective

60. Những việc làm tốt

- Phân vai Scrum rõ ràng: PO (Phúc), SM (Khang, Hiệp), Dev Team (Huy, Tùng).
- Tổ chức được nhiều buổi Daily Scrum (Sprint 2, 3) với biên bản tóm tắt đầy đủ.
- Khang hoàn thành nhiệm vụ nhanh nhất nhóm (trong vòng 2.6 tiếng sau khi được giao việc).
- Team phối hợp tốt giữa FE & BE trong quá trình phát triển tính năng.
- Xác định được blockers và action items cụ thể sau mỗi Daily Scrum.

61. Những việc chưa tốt

- API contract chưa thống nhất giữa FE & BE, gây trễ tiến độ integration.
- Naming convention chưa đồng bộ giữa các module.
- Một số thành viên bận lịch học, ảnh hưởng timeline Sprint.
- Thiếu môi trường staging cho integration tests.
- Chưa có sự giao tiếp, bàn luận thường xuyên giữa các thành viên ngoài Daily Scrum.

62. Đánh giá đóng góp từng thành viên

63. Phúc (Product Owner)

- Xác nhận requirements, acceptance criteria đầy đủ.
- Chuẩn bị UAT checklist và test cases.

64. Khang (Scrum Master)

- Hoàn thành nhiệm vụ sớm nhất nhóm.
- Hỗ trợ kiểm tra CI pipeline và code review.

65. Hiệp (Scrum Master)

- Theo dõi tiến độ integration và DevOps.
- Follow-up Firebase notification.

66. Huy (Developer)

- Hoàn thiện API CRUD, chấm điểm, xử lý conflict detection.
- Viết unit test và deploy staging.

67. Tùng (Developer)

- Thiết kế UI flow và tích hợp frontend.
- Triển khai notification UI và test upload.

68. Biện pháp cải tiến

- Thống nhất API contract và naming convention trước khi code.
- Thiết lập môi trường staging sớm cho integration tests.
- Tăng cường giao tiếp nhóm ngoài Daily Scrum.
- Cập nhật timeline Sprint linh hoạt theo lịch học của thành viên.
- Ghi nhận feedback để cải thiện cho Sprint tiếp theo.

69. So sánh Agile vs. Waterfall (Vũ Quốc Huy)

70. Phân tích Agile vs Waterfall lĩnh vực Rủi ro

71. Em xin lỗi vì đã hỏi cái máy ạ 🙄

72. Nếu thực hiện Waterfall thì sao?

Mô hình Waterfall tiếp cận dự án theo tư duy tuyến tính: *Yêu cầu -> Thiết kế -> Lập trình -> Kiểm thử -> Triển khai.*

Trong dự án EdTech (Công nghệ giáo dục), Waterfall mang lại những rủi ro đặc thù sau:

1. **Rủi ro “Cú nổ lớn” (Big Bang Risk) ở giai đoạn cuối:**
 - Trong Waterfall, việc kiểm thử (Testing) và chấp nhận sản phẩm (UAT) diễn ra ở cuối dự án. Nếu trung tâm ngoại ngữ phát hiện ra luồng điểm danh không phù hợp với thực tế vận hành vào tháng thứ 6, chi phí sửa chữa là cực kỳ lớn vì phải đập đi xây lại từ khâu thiết kế CSDL.
2. **Rủi ro về sự thay đổi yêu cầu (Requirements Creep):**
 - Ngành giáo dục thường xuyên thay đổi (ví dụ: thay đổi quy chế thi, hình thức học Online sang Hybrid). Waterfall yêu cầu “đóng băng” yêu cầu từ đầu. Nếu thị trường thay đổi trong lúc đang Code, sản phẩm khi ra mắt sẽ trở nên lỗi thời.
3. **Rủi ro về sự hiểu lầm (Communication Gap):**
 - Khách hàng (Trung tâm) thường không hình dung được giao diện qua tài liệu giấy. Họ chỉ nhận ra mình cần gì khi thấy phần mềm chạy thực tế. Waterfall trì hoãn việc “thấy” này quá lâu, dẫn đến rủi ro sản phẩm làm ra đúng **tài liệu** nhưng sai **nhu cầu**.
4. **Rủi ro tích hợp (Integration Risk):**
 - Việc để dành khâu tích hợp các module (Điểm danh + Video + Thanh toán) đến cuối cùng thường gây ra bùng nổ lỗi hệ thống nghiêm trọng sát ngày Deadline.

73. Nếu thực hiện Agile thì sao?

Mô hình Agile (ví dụ: Scrum) tiếp cận theo hướng lặp và tăng trưởng: Chia nhỏ dự án thành các Sprint (2-4 tuần), mỗi Sprint cho ra một phần mềm chạy được.

Dù linh hoạt, Agile không miễn nhiễm với rủi ro, đặc biệt là các rủi ro về quản lý:

1. Rủi ro về phạm vi (Scope Creep) không kiểm soát:

- Vì Agile chào đón sự thay đổi, Trung tâm ngoại ngữ có thể liên tục yêu cầu thêm tính năng mới (ví dụ: thêm Gamification, AI chấm điểm) khiến dự án không bao giờ kết thúc và ngân sách bị phình to.

2. Rủi ro thiếu tài liệu (Documentation Debt):

- Agile ưu tiên phần mềm chạy được hơn tài liệu. Nếu đội ngũ phát triển không kỷ luật, khi bàn giao dự án cho đội IT của trung tâm bảo trì, họ sẽ gặp khó khăn vì thiếu tài liệu kỹ thuật chi tiết.

3. Rủi ro phụ thuộc vào sự tham gia của khách hàng (User Engagement):

- Agile yêu cầu Product Owner (đại diện trung tâm) phải làm việc sát sao hàng ngày. Nếu họ bận rộn hoặc không đủ thẩm quyền quyết định, các Sprint sẽ bị tắc nghẽn, dẫn đến lãng phí nguồn lực team Dev.

4. Rủi ro về kiến trúc (Architectural Instability):

- Việc phát triển nhanh mún từng tính năng có thể dẫn đến một kiến trúc “chắp vá” nếu không có Tech Lead giỏi định hướng từ đầu, khiến hệ thống khó mở rộng sau này (Technical Debt).

74. Bảng so sánh Ưu và Nhược điểm

Dưới đây là bảng đối chiếu chi tiết dựa trên đặc thù của dự án Quản lý lớp học.

Mô hình Waterfall	Mô hình Agile
<p>Ưu điểm:</p> <ul style="list-style-type: none">• Dễ dự báo ngân sách: Giá và thời gian được chốt ngay từ đầu (Fixed Price). Phù hợp nếu trung tâm có ngân sách cứng.• Kỷ luật cao: Quy trình rõ ràng, tài liệu đầy đủ. Dễ dàng chuyển giao nhân sự.• Cấu trúc ổn định: Thiết kế tổng thể (System Architecture) được làm kỹ từ đầu, hệ thống chặt chẽ.	<p>Ưu điểm:</p> <ul style="list-style-type: none">• Giảm rủi ro sai sản phẩm: Khách hàng xem Demo mỗi 2 tuần, sửa sai ngay lập tức.• Time-to-market sớm: Ra mắt bản MVP (chỉ có xếp lịch & vào học) sau 1-2 tháng để dùng ngay.• Chất lượng cao: Testing liên tục trong từng Sprint, lỗi được xử lý sớm.
<p>Nhược điểm:</p> <ul style="list-style-type: none">• Độ trễ phản hồi: Khách hàng phải đợi rất lâu mới thấy sản phẩm thực tế.• Rủi ro thất bại cao: Nếu phân tích sai nghiệp vụ lỗi (ví dụ: cách tính lương GV), dự án có thể sụp đổ.• Cứng nhắc: Rất khó và tốn kém để thay đổi tính năng khi đã vào giai đoạn Code.	<p>Nhược điểm:</p> <ul style="list-style-type: none">• Khó dự báo ngân sách: Tổng chi phí biến động nếu yêu cầu thay đổi nhiều.• Đòi hỏi nhân sự cao: Team cần người tự giác, đa năng (Cross-functional), giao tiếp tốt.• Căng thẳng: Áp lực trả hàng liên tục mỗi Sprint dễ gây mệt mỏi (Burnout).

Bảng 1: So sánh chi tiết Waterfall và Agile trong dự án LMS

75. Ví dụ minh họa (Case Study)

Dự án: Ứng dụng Quản lý Lớp học cho Trung tâm Anh ngữ ABC.

76. Kịch bản 1: Áp dụng Waterfall (Kịch bản thất bại điển hình)

- **Tháng 1-2:** Team dành 2 tháng viết tài liệu đặc tả dày 200 trang. Quy định “Học viên vắng 3 buổi sẽ bị khóa tài khoản”.
- **Tháng 3-5:** Lập trình viên code miệt mài (Coding phase). Không có giao tiếp với trung tâm.

- **Tháng 6 (UAT):** Bàn giao. Giám đốc trung tâm dùng thử và thốt lên: *“Thực tế là học viên vẫn nhưng có phép thì không được khóa. Và giao diện này khó dùng quá, phụ huynh lớn tuổi không biết bấm vào đâu để xem điểm.”*
- **Hậu quả:** Phải sửa lại Logic Core và viết lại UI. Dự án chậm tiến độ 3 tháng, chi phí tăng 40%.

77. Kịch bản 2: Áp dụng Agile (Kịch bản thành công)

- **Sprint 1 (2 tuần):** Làm tính năng “Danh sách lớp” và “Điểm danh cơ bản”.
- **Review Sprint 1:** Giáo vụ dùng thử và phản hồi: *“Nút điểm danh trên điện thoại bé quá, khó bấm”*. Team sửa ngay trong ngày hôm sau.
- **Sprint 2:** Làm tính năng “Học Online qua Zoom”.
- **Review Sprint 2:** Trung tâm nhận thấy API của Zoom bị giới hạn 40 phút. Họ yêu cầu chuyển sang tích hợp Google Meet. -> **Xử lý rủi ro:** Vì chưa làm quá sâu, team chuyển hướng sang Google Meet ngay lập tức mà không đập bỏ quá nhiều code cũ.
- **Kết quả:** Sau 3 tháng, trung tâm đã có một bản App dùng tốt cho việc học và điểm danh, dù tính năng Báo cáo chưa xong nhưng App đã sinh ra giá trị.

78. Nhận xét mô hình hợp lý nhất

Dựa trên phân tích rủi ro và đặc thù của dự án “Ứng dụng quản lý lớp học”, tôi đề xuất:

Mô hình được chọn: Agile (Cụ thể là Scrum kết hợp Kanban)

79. Lý do:

1. **Tính chất “Con người” của dự án giáo dục:** Ứng dụng giáo dục phục vụ học viên, giáo viên và phụ huynh. Trải nghiệm người dùng (UX) là yếu tố sống còn. Waterfall không cho phép thử nghiệm UX sớm. Agile cho phép người dùng thật trải nghiệm sớm và tinh chỉnh giao diện cho phù hợp với hành vi của họ.
2. **Quản trị rủi ro biến động nghiệp vụ:** Các trung tâm ngoại ngữ thường xuyên thay đổi chương trình khuyến mãi, cách xếp lớp hoặc quy chế để cạnh tranh. Agile cho phép thay đổi Backlog linh hoạt để thích nghi với chiến lược kinh doanh của trung tâm.
3. **Kiểm soát chất lượng liên tục:** Hệ thống LMS cần độ ổn định cao (không được sập khi đang học). Agile bắt buộc testing liên tục (Continuous

Integration/Continuous Testing), giúp đảm bảo hệ thống ổn định hơn so với việc dồn testing vào cuối dự án như Waterfall.

4. **Tối ưu dòng tiền (ROI):** Với Agile, trung tâm có thể triển khai module “Tuyển sinh” trước để thu hút học viên mới ngay trong tháng thứ 2, tạo ra dòng tiền để nuôi dự án tiếp tục phát triển các module sau.

Kết luận: Waterfall quá rủi ro cho các dự án phần mềm hiện đại hướng tới người dùng cuối (B2C/B2B2C). Agile tuy đòi hỏi sự cam kết cao hơn về giao tiếp nhưng mang lại sự an toàn về mặt giá trị sản phẩm (Value-driven delivery).

80. Báo cáo cuối kì (Đỗ Đình Khang)

Quản lý dự án Phát triển Phần mềm - SE358.Q11

Đề tài 2: Ứng dụng quản lý lớp học trực tuyến cho trung tâm ngoại ngữ

Báo cáo cuối kì

02.12.2025

Nhóm 2

Contents

1. Sprint Backlog (Bùi Văn Tùng)	1
2. Sprint Backlog	1
3. Phân vai Scrum	1
4. Sản phẩm chủ sở hữu (Product Owner)	1
5. Chủ nhân xì trum (Scrum Master)	1
6. Biệt đội phát triển (Development Team)	1
7. Sprint 2: Quản lý lớp học và lịch học	2
8. Sprint Goal	2
9. User Stories	2
10. User Story 1: Giảng viên tạo lớp mới	2
11. User Story 2: Giảng viên đặt lịch học	2
12. User Story 3: Học viên xem danh sách lớp	2
13. User Story 4: Giảng viên phân công giảng viên	3
14. User Story 5: API validation cho lịch học	3
15. User Story 6: Conflict detection cho lịch giảng viên	3
16. User Story 7: Integration class & user profile	3
17. Velocity Sprint 2	4
18. Blockers và Action Items	4
19. Blocker 1: Conflict detection policy chưa rõ	4
20. Blocker 2: API contract chưa thống nhất	4
21. Blocker 3: Authorization chưa rõ	5
22. Blocker 4: Thiếu môi trường staging	5

23. Summary Action Items Sprint 2	5
24. Sprint 3: Quản lý bài tập và chấm điểm	6
25. Sprint Goal	6
26. User Stories	6
27. User Story 1: Giảng viên tạo bài tập	6
28. User Story 2: Giảng viên set deadline	6
29. User Story 3: Học viên xem danh sách bài tập	6
30. User Story 4: Học viên nộp bài	7
31. User Story 5: Giảng viên xem bài nộp	7
32. User Story 6: Giảng viên chấm điểm và comment	7
33. User Story 7: Học viên nhận thông báo điểm	7
34. User Story 8: API validation file upload	8
35. Velocity Sprint 3	8
36. Blockers và Action Items	8
37. Blocker 1: Firebase notification chưa setup	8
38. Blocker 2: API response format chưa thống nhất	8
39. Blocker 3: Naming convention chưa đồng bộ	9
40. Blocker 4: Team member lịch học bận	9
41. Summary Action Items Sprint 3	9
42. Tóm tắt Metrics	10
43. Velocity Trends	10
44. Key Lessons	10
45. Recommendations	10

46. Biên bản Daily Scum 1 (Tạ Hoàng Hiệp)	11
47. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 2)	11
48. Làm gì hôm qua?	11
49. Bữa này gì làm?	11
50. Mặc gì vương cái?	12
51. Tắt bản biên tóm	12
52. Biên bản Daily Scum 2 (Tạ Hoàng Hiệp)	12
53. Mini-Sprint Scrum — Biên bản tóm tắt (Sprint 3)	12
54. Làm gì hôm qua?	12
55. Bữa này gì làm?	13
56. Mặc gì vương cái?	13
57. Tắt bản biên tóm	13
58. Sprint Retrospective (Trương Hoàng Phúc)	14
59. Prompt Cho AI	14
60. Những việc làm tốt	14
61. Những việc chưa tốt	14
62. Đánh giá đóng góp từng thành viên	14
63. Phúc (Product Owner)	14
64. Khang (Scrum Master)	15
65. Hiệp (Scrum Master)	15
66. Huy (Developer)	15
67. Tùng (Developer)	15
68. Biện pháp cải tiến	15

69. So sánh Agile vs. Waterfall (Vũ Quốc Huy)	16
70. Phân tích Agile vs Waterfall lĩnh vực Rủi ro	16
71. Em xin lỗi vì đã hỏi cái máy ạ 🙄	16
72. Nếu thực hiện Waterfall thì sao?	16
73. Nếu thực hiện Agile thì sao?	16
74. Bảng so sánh Ưu và Nhược điểm	17
75. Ví dụ minh họa (Case Study)	18
76. Kịch bản 1: Áp dụng Waterfall (Kịch bản thất bại điển hình)	18
77. Kịch bản 2: Áp dụng Agile (Kịch bản thành công)	19
78. Nhận xét mô hình hợp lý nhất	19
79. Lý do:	19
80. Báo cáo cuối kì (Đỗ Đình Khang)	20
81. Giới thiệu dự án	1
81.1. Bối cảnh dự án	1
81.2. Mục tiêu dự án	1
81.3. Phạm vi dự án	1
81.3.1. Các chức năng chính	1
81.3.2. Ranh giới dự án	2
81.4. Stakeholders chính	2
81.5. Lợi ích kỳ vọng	3
82. Tóm tắt tài liệu quản lý	3
82.1. Project Charter (Hiển chương dự án)	3
82.1.1. Thông tin cơ bản	4
82.1.2. Mục đích dự án	4
82.1.3. Sản phẩm bàn giao chính	4
82.1.4. Các mốc quan trọng	4
82.1.5. Ngân sách	4
82.1.6. Quyền hạn Project Manager	5
82.1.7. Rủi ro cấp cao	5
82.2. Work Breakdown Structure (WBS)	5
82.2.1. Giai đoạn 1: Phân tích yêu cầu	5
82.2.2. Giai đoạn 2: Thiết kế hệ thống	5

82.2.3. Giai đoạn 3: Phát triển và Kiểm thử	6
82.2.4. Giai đoạn 4: Viết báo cáo và Thuyết trình	6
82.2.5. Quản lý rủi ro trong WBS	7
82.3. Quản lý Stakeholder	7
82.3.1. Phân loại Stakeholder	7
82.3.2. Chiến lược giao tiếp	8
82.4. Quản lý Chi phí	8
82.4.1. Ước lượng chi phí	8
82.4.2. Baseline chi phí	10
82.4.3. Kiểm soát chi phí	10
83. Phân tích so sánh Agile vs Waterfall	10
83.1. Tổng quan về hai phương pháp	10
83.1.1. Waterfall (Thác nước)	10
83.1.2. Agile (Linh hoạt)	11
83.2. So sánh chi tiết	11
83.3. Áp dụng cho dự án quản lý lớp học	11
83.3.1. Phân tích bối cảnh dự án	11
83.3.2. Đề xuất: Hybrid Approach (Kết hợp)	12
83.3.3. Lợi ích của Hybrid Approach	12
84. Kết quả mô phỏng Sprint	12
84.1. Tổng quan Sprint Planning	12
84.2. Sprint 3: Quản lý bài tập và chấm điểm	13
84.2.1. Sprint Goal	13
84.2.2. Sprint Backlog	13
84.2.3. Sprint Planning Meeting (Ngày 1 Sprint)	13
84.2.4. Daily Standup (Ngày 5 Sprint)	14
84.2.5. Sprint Review (Ngày 14 Sprint - cuối Sprint)	14
84.2.6. Sprint Retrospective (Ngày 14 Sprint - sau Review)	15
84.3. Metrics và Hiệu suất Sprint	15
84.3.1. Burndown Chart	15
84.3.2. Velocity Tracking	15
84.4. Bài học từ Sprint Simulation	16
84.4.1. Về Planning	16
84.4.2. Về Execution	16
84.4.3. Về Collaboration	16
85. Bài học	16
85.1. Về Quản lý Dự án	16
85.1.1. Lập kế hoạch chi tiết từ đầu	16
85.1.2. Quản lý Stakeholder hiệu quả	17
85.1.3. Kiểm soát Scope Creep	17

85.2. Về Phát triển Phần mềm	17
85.2.1. Thiết kế Database từ đầu	17
85.2.2. Code Review và Testing	17
85.2.3. Documentation	18
85.3. Về Teamwork	18
85.3.1. Giao tiếp trong Team	18
85.3.2. Phân công công việc	18
85.3.3. Continuous Improvement	18
85.4. Về Công nghệ	19
85.4.1. Lựa chọn Tech Stack	19
85.4.2. Cloud và Deployment	19
85.5. Khuyến nghị cho Dự án Tương lai	19
85.6. Kết luận	19
86. Tài liệu tham khảo	21
87. Retrospective C	22
87.1. Đánh giá trải nghiệm Agile	22
87.2. Những khác biệt so với Waterfall	22
87.3. Những việc làm tốt	22
87.4. Những việc làm chưa tốt	22
87.5. Đánh giá đóng góp từng thành viên	22
87.5.1. Hiệp	22
87.5.2. Khang	23
87.5.3. Huy	23
87.5.4. Phúc	23
87.5.5. Tùng	23
87.6. Bài học rút ra	23
87.7. Đề xuất cải thiện	23

81. Giới thiệu dự án

81.1. Bối cảnh dự án

Trong bối cảnh công nghệ số hóa đang phát triển mạnh mẽ, các trung tâm ngoại ngữ đang đối mặt với nhiều thách thức trong việc quản lý lớp học, giảng viên và học viên một cách hiệu quả. Việc sử dụng các phương pháp quản lý truyền thống như sổ sách giấy tờ, bảng Excel riêng lẻ không chỉ tốn thời gian mà còn dễ xảy ra sai sót, khó đồng bộ thông tin giữa các bộ phận.

Nhận thấy nhu cầu cấp thiết này, dự án “Ứng dụng quản lý lớp học trực tuyến cho trung tâm ngoại ngữ” ra đời nhằm xây dựng một hệ thống tổng thể, giúp số hóa toàn bộ quy trình quản lý giáo dục tại trung tâm. Ứng dụng không chỉ hỗ trợ quản trị viên trong việc theo dõi lớp học, giảng viên, học viên mà còn tạo ra môi trường học tập trực tuyến thuận tiện, linh hoạt cho người học.

81.2. Mục tiêu dự án

Dự án được xây dựng với các mục tiêu cụ thể như sau:

- Về phạm vi (Scope): Xây dựng ứng dụng mobile quản lý lớp học trực tuyến với đầy đủ các chức năng: quản lý lớp học, giảng viên, học viên, điểm số, lịch học và tài liệu học tập. Tất cả các module chính phải hoạt động ổn định và được nghiệm thu thành công.
- Về thời gian (Time): Hoàn thành toàn bộ dự án trong vòng 4-6 tháng kể từ khi khởi động, đảm bảo 100% các mốc quan trọng (milestones) được hoàn thành đúng hoặc sớm hơn tiến độ đã định.
- Về chi phí (Cost): Triển khai dự án trong phạm vi ngân sách được phê duyệt là 200 triệu VNĐ, với tổng chi phí thực tế không vượt quá $\pm 10\%$ ngân sách dự toán.
- Về chất lượng (Quality): Đảm bảo hệ thống có tính bảo mật cao, không để lộ dữ liệu người dùng; giao diện thân thiện, dễ sử dụng và có khả năng mở rộng trong tương lai. Mục tiêu đạt được mức độ hài lòng tối thiểu 70% từ người dùng cuối.

81.3. Phạm vi dự án

81.3.1. Các chức năng chính

Ứng dụng cung cấp các tính năng cốt lõi sau:

- Đối với Quản trị viên:
 - Quản lý thông tin giảng viên và học viên (thêm, sửa, xóa, tìm kiếm)
 - Quản lý lớp học: tạo lớp, phân công giảng viên, xếp lịch học
 - Theo dõi tình trạng hoạt động của các lớp học
 - Xem báo cáo và thống kê tổng hợp về số lượng học viên, lớp học, doanh thu
- Đối với Giảng viên:
 - Quản lý thông tin lớp học được phân công
 - Tạo và quản lý bài giảng, tài liệu học tập
 - Điểm danh học viên theo từng buổi học
 - Chấm điểm bài tập, bài kiểm tra và quản lý điểm số
 - Tương tác với học viên qua hệ thống
- Đối với Học viên:
 - Đăng ký tham gia các lớp học
 - Xem lịch học, thông tin giảng viên
 - Truy cập tài liệu học tập, bài giảng
 - Nộp bài tập trực tuyến
 - Xem điểm số và nhận phản hồi từ giảng viên
 - Học trực tuyến thông qua các bài giảng số

81.3.2. Ranh giới dự án

Để đảm bảo dự án hoàn thành đúng tiến độ và ngân sách, các yếu tố sau được xác định rõ ràng:

- Nằm trong phạm vi:
 - Phát triển ứng dụng mobile (Android/iOS)
 - Hệ thống backend API và cơ sở dữ liệu
 - Các tính năng quản lý cơ bản như đã liệt kê
 - Đào tạo người dùng về cách sử dụng hệ thống
- Nằm ngoài phạm vi:
 - Hệ thống thanh toán trực tuyến (sẽ được cân nhắc trong giai đoạn sau)
 - Tích hợp với các nền tảng mạng xã hội
 - Hỗ trợ các ngành học khác ngoài ngoại ngữ
 - Marketing và quảng bá ứng dụng
 - Tích hợp AI/ML cho gợi ý học tập cá nhân hóa

81.4. Stakeholders chính

Dự án có sự tham gia của nhiều bên liên quan với các vai trò và mức độ ảnh hưởng khác nhau:

- Nhà tài trợ (Sponsor): Ban Giám đốc Trung tâm Ngoại ngữ ABC, đại diện bởi ông Nguyễn Văn An, có quyền lực và mức độ quan tâm cao nhất đối với

dự án. Sponsor chịu trách nhiệm phê duyệt ngân sách, theo dõi tiến độ và đảm bảo dự án đi đúng mục tiêu kinh doanh.

- Quản lý dự án (Project Manager): Bà Lê Thị Hương, người chịu trách nhiệm chính trong việc quản lý phạm vi, tiến độ, chất lượng và ngân sách của toàn bộ dự án. PM duy trì giao tiếp thường xuyên với tất cả các bên liên quan.
- Nhóm phát triển: Bao gồm chuyên viên phân tích nghiệp vụ (BA) - Phạm Minh Tuấn, trưởng nhóm phát triển - Trần Thảo Nhi, cùng các lập trình viên, nhà thiết kế và kỹ sư hệ thống.
- Nhóm kiểm thử: Do ông Nguyễn Văn Bình làm trưởng nhóm, chịu trách nhiệm đảm bảo chất lượng sản phẩm.
- Người dùng cuối: Bao gồm giảng viên (là chuyên gia nghiệp vụ cung cấp yêu cầu) và học viên (người dùng chính của ứng dụng).
- Các bộ phận hỗ trợ: Bộ phận IT hạ tầng, phòng Marketing, và có thể có sự giám sát từ Sở Giáo dục & Đào tạo về mặt tuân thủ quy định.

81.5. Lợi ích kỳ vọng

Dự án mong đợi mang lại các lợi ích thiết thực:

- Đối với trung tâm:
 - Tự động hóa quy trình quản lý, tiết kiệm thời gian và nhân lực
 - Giảm thiểu sai sót trong việc quản lý thông tin
 - Nâng cao hình ảnh chuyên nghiệp, hiện đại của trung tâm
 - Có cơ sở dữ liệu tập trung để phân tích, ra quyết định kinh doanh
- Đối với giảng viên:
 - Quản lý lớp học, học viên một cách thuận tiện
 - Tiết kiệm thời gian trong việc điểm danh, chấm điểm
 - Dễ dàng chia sẻ tài liệu và tương tác với học viên
- Đối với học viên:
 - Truy cập thông tin lớp học, lịch học mọi lúc mọi nơi
 - Học tập linh hoạt với tài liệu số
 - Nhận phản hồi kịp thời về kết quả học tập

82. Tóm tắt tài liệu quản lý

82.1. Project Charter (Hiển chương dự án)

Project Charter là văn bản chính thức khởi động dự án, được phê duyệt bởi Sponsor và trao quyền cho Project Manager điều hành dự án.

82.1.1. Thông tin cơ bản

- Tên dự án: Ứng dụng quản lý lớp học trực tuyến cho trung tâm ngoại ngữ
- Sponsor: Ban Giám đốc Trung tâm Ngoại ngữ ABC
- Project Manager: Nhóm 2
- Ngày chuẩn bị: 01/10/2025
- Khách hàng: Trung tâm Ngoại ngữ ABC và học viên

82.1.2. Mục đích dự án

Tạo ra một hệ thống trực tuyến giúp trung tâm ngoại ngữ quản lý lớp học, giảng viên và học viên hiệu quả hơn; đồng thời cung cấp nền tảng học tập linh hoạt, tiết kiệm thời gian và chi phí vận hành.

82.1.3. Sản phẩm bàn giao chính

1. Ứng dụng mobile hoàn chỉnh (Android/iOS)
2. Cơ sở dữ liệu tập trung về học viên, giảng viên, lớp học
3. Module quản lý điểm số, lịch học, tài liệu học tập
4. Hệ thống báo cáo và thống kê cho quản trị viên
5. Tài liệu hướng dẫn sử dụng và tài liệu kỹ thuật

82.1.4. Các mốc quan trọng

Milestone	Thời gian
Khởi động dự án	2 tuần
Phân tích yêu cầu	4 tuần
Thiết kế hệ thống	8 tuần
Phát triển và kiểm thử	8 tuần
Triển khai và đào tạo	2 tuần

82.1.5. Ngân sách

Tổng ngân sách được phê duyệt: **200 triệu VNĐ**

Ngân sách này bao gồm:

- Chi phí nhân lực (lập trình viên, designer, tester)
- Chi phí công nghệ (hosting, domain, license)
- Chi phí đào tạo và tài liệu
- Dự phòng 10% cho các rủi ro

82.1.6. Quyền hạn Project Manager

- Toàn quyền điều phối nhóm dự án
- Quyết định về staffing (nhân sự)
- Quản lý ngân sách và báo cáo cho Sponsor
- Giải quyết xung đột nội bộ nhóm
- Đưa ra quyết định kỹ thuật với sự tham vấn từ nhóm

82.1.7. Rủi ro cấp cao

- Rủi ro công nghệ: Hạ tầng server, vấn đề bảo mật dữ liệu
- Rủi ro người dùng: Chưa quen với hệ thống mới, cần thời gian làm quen
- Rủi ro nguồn lực: Hạn chế về ngân sách và thời gian triển khai
- Rủi ro scope creep: Yêu cầu thay đổi, mở rộng phạm vi ngoài kế hoạch

82.2. Work Breakdown Structure (WBS)

WBS là cấu trúc phân rã công việc, chia nhỏ dự án thành các gói công việc (work packages) có thể quản lý được. Dự án được tổ chức thành 4 giai đoạn chính:

82.2.1. Giai đoạn 1: Phân tích yêu cầu

- Thu thập yêu cầu:
- Tìm hiểu mô hình quản lý lớp học trực tuyến hiện đại
- Nghiên cứu các hệ thống tương tự (Google Classroom, Moodle, Canvas)
- Phỏng vấn stakeholders để thu thập nhu cầu thực tế
- Phân tích hệ thống:
- Xác định các chức năng chính: tạo lớp, quản lý học viên, tài liệu, bài tập, điểm số
- Vẽ sơ đồ use case, user stories
- Đặc tả yêu cầu chức năng và phi chức năng
- Xây dựng tài liệu SRS (Software Requirements Specification)

82.2.2. Giai đoạn 2: Thiết kế hệ thống

- Thiết kế giao diện (UI/UX):
- Wireframe và mockup cho các màn hình chính
- Giao diện đăng ký, đăng nhập, đăng xuất
- Giao diện trang chủ và dashboard
- Giao diện quản lý lớp học (tài liệu, bài tập, điểm)

- Giao diện quản trị hệ thống
- Thiết kế kỹ thuật:
- Thiết kế cơ sở dữ liệu (ERD): bảng user, class, assignment, grade, document
- Thiết kế API RESTful: authentication, class management, assignment submission, grading
- Thiết kế kiến trúc hệ thống: client-server, microservices (nếu cần)
- Thiết kế bảo mật: phân quyền (RBAC), mã hóa dữ liệu, session management

82.2.3. Giai đoạn 3: Phát triển và Kiểm thử

- Phát triển Front-end:
- Màn hình đăng nhập / đăng ký
- Trang chủ và navigation
- Màn hình quản lý lớp học
- Màn hình nộp bài và xem điểm
- Màn hình quản trị (admin panel)
- Phát triển Back-end:
- API quản lý người dùng (đăng ký, đăng nhập, phân quyền)
- API quản lý lớp học (CRUD operations)
- API quản lý bài tập và nộp bài
- API chấm điểm và phản hồi
- API báo cáo và thống kê
- Kiểm thử hệ thống:
- Kiểm thử chức năng (Functional Testing): đảm bảo tất cả tính năng hoạt động đúng
- Kiểm thử hiệu năng (Performance Testing): đánh giá khả năng chịu tải
- Kiểm thử bảo mật (Security Testing): phát hiện lỗ hổng bảo mật
- Kiểm thử tích hợp (Integration Testing)
- UAT (User Acceptance Testing) với người dùng thực

82.2.4. Giai đoạn 4: Viết báo cáo và Thuyết trình

- Viết báo cáo dự án:
- Mô tả yêu cầu và phạm vi hệ thống
- Trình bày thiết kế kiến trúc, cơ sở dữ liệu, giao diện
- Tổng hợp kết quả phát triển và kiểm thử
- Phân tích các thách thức và bài học kinh nghiệm
- Chuẩn bị thuyết trình:
- Xây dựng slide thuyết trình (giới thiệu, chức năng, demo)
- Chuẩn bị video demo hệ thống
- Phân công vai trò thuyết trình cho các thành viên

82.2.5. Quản lý rủi ro trong WBS

- **Scope Creep** (Mở rộng phạm vi ngoài kế hoạch):
- *Mô tả*: Stakeholders đề xuất thêm tính năng mới (AI, video conferencing) ngoài phạm vi ban đầu
- *Hậu quả*: Vượt thời gian, ngân sách, có thể không hoàn thành đúng hạn
- *Giải pháp*: Xác định phạm vi rõ ràng trong SRS, chỉ chấp nhận thay đổi sau đánh giá tác động
- **Hiểu sai yêu cầu**:
- *Mô tả*: Nhóm phát triển hiểu sai nhu cầu người dùng, xây dựng giao diện phức tạp khi họ chỉ cần thao tác đơn giản
- *Hậu quả*: Sản phẩm không đáp ứng kỳ vọng, phải làm lại
- *Giải pháp*: Thảo luận nhiều lần với stakeholders, xác nhận yêu cầu bằng prototype
- **Yêu cầu thay đổi trong triển khai**:
- *Mô tả*: Giảng viên yêu cầu thêm chức năng upload video khi hệ thống đang được xây dựng
- *Hậu quả*: Lịch trình bị kéo dài
- *Giải pháp*: Áp dụng quy trình Change Request, đánh giá tác động trước khi chấp nhận

82.3. Quản lý Stakeholder

Việc quản lý stakeholder đóng vai trò quan trọng trong thành công của dự án. Mỗi stakeholder có mức độ quyền lực, ảnh hưởng và quan tâm khác nhau.

82.3.1. Phân loại Stakeholder

Tên	Vai trò	Quyền lực	Quan tâm
Nguyễn Văn An	Sponsor	Mạnh	Cao
Lê Thị Hương	PM	Mạnh	Cao
Phạm Minh Tuấn	BA	Trung bình	Cao
Nhóm Giáo viên	Chuyên gia nghiệp vụ	Trung bình	Cao
Trần Thảo Nhi	Dev Lead	Mạnh	Cao
Nhóm Học viên	User	Thấp	Cao
Bộ phận IT	Hạ tầng	Trung bình	Trung bình
Nguyễn Văn Bình	QA Lead	Trung bình	Cao

82.3.2. Chiến lược giao tiếp

- Với Sponsor (Nguyễn Văn An):
 - Họp báo cáo hàng tuần về tiến độ, ngân sách
 - Gửi email cập nhật các vấn đề quan trọng
 - Tham vấn khi có quyết định lớn về scope, budget
- Với Nhóm phát triển:
 - Daily standup meeting (15 phút mỗi ngày)
 - Sprint planning, review, retrospective (nếu dùng Agile)
 - Slack/Teams cho giao tiếp nhanh
- Với Giảng viên (Chuyên gia nghiệp vụ):
 - Workshop hàng tuần để thu thập feedback
 - Demo sản phẩm định kỳ để xác nhận yêu cầu
 - Khảo sát mức độ hài lòng
- Với Học viên:
 - Khảo sát trực tuyến về trải nghiệm sử dụng
 - Biểu mẫu phản hồi trong app
 - Phiên beta testing với nhóm người dùng đại diện

82.4. Quản lý Chi phí

82.4.1. Ước lượng chi phí

Dự án sử dụng nhiều phương pháp ước lượng chi phí để đảm bảo độ chính xác:

- Parametric Estimates (Ước lượng tham số):
 - Dev extra hours cho rework: $1,500 \text{ giờ} \times 200,000 \text{ đ/giờ} = 300,000,000 \text{ đ}$
 - QA overrun hours: $150 \text{ giờ} \times 150,000 \text{ đ/giờ} = 22,500,000 \text{ đ}$
 - Cloud hosting overrun: $12 \text{ instance-months} \times 1,500,000 \text{ đ} = 18,000,000 \text{ đ}$
 - 3rd-party licenses: $3 \times 5,000,000 \text{ đ} = 15,000,000 \text{ đ}$
 - Security audit: $1 \times 20,000,000 \text{ đ} = 20,000,000 \text{ đ}$
 - Data migration: $80 \text{ giờ} \times 150,000 \text{ đ/giờ} = 12,000,000 \text{ đ}$
 - Training sessions: $4 \text{ sessions} \times 10,000,000 \text{ đ} = 40,000,000 \text{ đ}$
 - Legal/compliance: $1 \times 8,000,000 \text{ đ} = 8,000,000 \text{ đ}$
 - UX adjustments: $60 \text{ giờ} \times 180,000 \text{ đ/giờ} = 10,800,000 \text{ đ}$
 - *Subtotal*: 446,300,000 đ
 - *Contingency (10%)*: 44,630,000 đ
 - **Grand Total: 490,930,000 đ (làm tròn: 491 triệu đ)**

Dựa vào các dự án tương tự đã thực hiện:

- LMS nhỏ (2019): $300 \text{ triệu} \times 1.4 = 420 \text{ triệu}$
- Website quản lý lớp (2020): $200 \text{ triệu} \times 1.25 = 250 \text{ triệu}$
- App học tiếng (2021): $150 \text{ triệu} \times 1.5 = 225 \text{ triệu}$

- Hạ tầng cloud (2022): $90 \text{ triệu} \times 1.6 = 144 \text{ triệu}$
- Tích hợp thanh toán (2020): $50 \text{ triệu} \times 1.2 = 60 \text{ triệu}$
- Audit bảo mật (2023): $35 \text{ triệu} \times 1.3 = 45.5 \text{ triệu}$
- Đào tạo nhân sự (2021): $30 \text{ triệu} \times 1.5 = 45 \text{ triệu}$
- Thiết kế UX/UI (2022): $40 \text{ triệu} \times 1.2 = 48 \text{ triệu}$
- *Trung bình*: Khoảng **405 triệu đ**
- Three-Point Estimates (Ước lượng ba điểm):
Sử dụng công thức PERT: $\text{Expected} = (\text{Optimistic} + 4 \times \text{Most Likely} + \text{Pessimistic}) / 6$

Hạng mục	Optimistic	Most Likely	Expected
Dev rework	250,000,000đ	300,000,000đ	308,300,000đ
QA overrun	18,000,000đ	22,500,000đ	23,000,000đ
Cloud hosting	12,000,000đ	18,000,000đ	19,000,000đ
3rd-party licenses	10,000,000đ	15,000,000đ	15,800,000đ
Security audit	15,000,000đ	20,000,000đ	21,700,000đ
Data migration	8,000,000đ	12,000,000đ	12,700,000đ
Training	36,000,000đ	40,000,000đ	40,700,000đ
Legal	6,000,000đ	8,000,000đ	8,800,000đ
UX adjustments	8,000,000đ	10,800,000đ	11,500,000đ
Total			461 triệu

- Bottom-Up Estimates (Ước lượng từ dưới lên):
Chi tiết từng hạng mục:

ID	Hạng mục	Chi phí (VND)
1	Dev rework (1,500h × 200,000 đ)	302,000,000 đ
2	QA overrun (300h × 150,000 đ)	60,000,000 đ
3	Cloud hosting + setup	31,200,000 đ
4	3rd-party licenses	18,000,000 đ
5	Security audit	24,000,000 đ
6	Data migration (80h × 150,000 đ)	14,400,000 đ
7	Training (material + travel)	42,000,000 đ
8	Legal/compliance	10,800,000 đ
9	UX adjustments (60h × 180,000 đ)	12,960,000 đ
Total		513,060,000 đ

82.4.2. Baseline chi phí

Tổng ngân sách: **500 triệu VND**

Phân bổ:

- Nhân lực (65%): 325 triệu
- Công nghệ (20%): 100 triệu
- Đào tạo & tài liệu (5%): 25 triệu
- Dự phòng (10%): 20 triệu

82.4.3. Kiểm soát chi phí

- Control Thresholds:
 - $\leq 5\%$: PM tự điều chỉnh
 - 5-10%: Cần thông báo Sponsor
 - $> 10\%$: Bắt buộc Change Request
- Performance Measurement:
 - Áp dụng Earned Value Management (EVM)
 - Tính toán CPI (Cost Performance Index) và SPI (Schedule Performance Index)
 - Báo cáo hàng tháng về AC (Actual Cost), EV (Earned Value), PV (Planned Value)

83. Phân tích so sánh Agile vs Waterfall

83.1. Tổng quan về hai phương pháp

83.1.1. Waterfall (Thác nước)

Waterfall là phương pháp quản lý dự án tuần tự, theo từng giai đoạn rõ ràng: Yêu cầu → Thiết kế → Phát triển → Kiểm thử → Triển khai → Bảo trì. Mỗi giai đoạn phải hoàn thành trước khi chuyển sang giai đoạn tiếp theo.

- Ưu điểm:
 - Dễ hiểu, dễ quản lý với timeline rõ ràng
 - Tài liệu đầy đủ, chi tiết ở mỗi giai đoạn
 - Phù hợp với dự án có yêu cầu ổn định, ít thay đổi
 - Dễ ước lượng chi phí, thời gian từ đầu
- Nhược điểm:
 - Kém linh hoạt, khó thay đổi yêu cầu giữa chừng
 - Phát hiện lỗi muộn (thường ở giai đoạn testing)

- Rủi ro cao nếu hiểu sai yêu cầu từ đầu
- Khách hàng chỉ thấy sản phẩm ở giai đoạn cuối

83.1.2. Agile (Linh hoạt)

Agile là phương pháp lặp đi lặp lại (iterative), chia dự án thành các Sprint ngắn (1-4 tuần). Mỗi Sprint tạo ra một phần sản phẩm có thể sử dụng được.

- Ưu điểm:
 - Linh hoạt, dễ thích nghi với thay đổi
 - Phản hồi nhanh từ khách hàng mỗi Sprint
 - Phát hiện và sửa lỗi sớm
 - Động lực cao cho team nhờ kết quả thường xuyên
 - Giảm rủi ro nhờ kiểm tra liên tục
- Nhược điểm:
 - Khó ước lượng tổng thời gian, chi phí từ đầu
 - Yêu cầu sự tham gia tích cực của khách hàng
 - Tài liệu có thể không đầy đủ bằng Waterfall
 - Cần team có kinh nghiệm, kỷ luật cao

83.2. So sánh chi tiết

Tiêu chí	Waterfall	Agile
Cấu trúc	Tuần tự, từng giai đoạn	Lặp lại, nhiều Sprint
Linh hoạt	Thấp	Cao
Tài liệu	Đầy đủ, chi tiết	Vừa đủ, cập nhật liên tục
Phản hồi KH	Cuối dự án	Mỗi Sprint (1-4 tuần)
Rủi ro	Cao nếu sai từ đầu	Thấp, phát hiện sớm
Ước lượng	Dễ từ đầu	Khó, cập nhật mỗi Sprint
Team size	Lớn, phân tách rõ	Nhỏ, cross-functional
Phù hợp	Yêu cầu rõ ràng, ổn định	Yêu cầu thay đổi nhiều

83.3. Áp dụng cho dự án quản lý lớp học

83.3.1. Phân tích bối cảnh dự án

Dự án “Ứng dụng quản lý lớp học trực tuyến” có những đặc điểm sau:

- Yêu cầu khá rõ ràng từ đầu (các chức năng quản lý cơ bản)

- Có thể có thay đổi nhỏ dựa trên feedback từ giáo viên, học viên
- Cần demo sớm để lấy ý kiến người dùng
- Thời gian và ngân sách hạn chế (4-6 tháng, 500 triệu)
- Team nhỏ, cần sự phối hợp chặt chẽ

83.3.2. Đề xuất: Hybrid Approach (Kết hợp)

Sau khi phân tích, nhóm đề xuất áp dụng **phương pháp kết hợp** giữa Waterfall và Agile:

Giai đoạn đầu (Phân tích, Thiết kế): Waterfall

- Phân tích yêu cầu kỹ lưỡng, tạo tài liệu SRS đầy đủ
- Thiết kế kiến trúc, cơ sở dữ liệu, API chi tiết
- Lý do: Cần nền tảng vững chắc, tránh thiết kế lại nhiều lần

Giai đoạn phát triển (Development): Agile

- Chia thành các Sprint 2 tuần
- Mỗi Sprint phát triển 1-2 tính năng hoàn chỉnh
- Daily standup, Sprint review với stakeholders
- Lý do: Linh hoạt với feedback, phát hiện lỗi sớm

Giai đoạn triển khai (Deployment): Waterfall

- Kiểm thử tổng thể (UAT)
- Đào tạo người dùng theo kế hoạch
- Triển khai chính thức
- Lý do: Cần quy trình rõ ràng, đảm bảo chất lượng

83.3.3. Lợi ích của Hybrid Approach

- Kết hợp tính ổn định của Waterfall với sự linh hoạt của Agile
- Giảm rủi ro thiết kế sai nhờ phân tích kỹ đầu
- Tận dụng feedback sớm trong giai đoạn phát triển
- Phù hợp với team nhỏ và stakeholders có kinh nghiệm hạn chế về Agile thuần túy

84. Kết quả mô phỏng Sprint

84.1. Tổng quan Sprint Planning

Trong giai đoạn phát triển, nhóm áp dụng Scrum framework với các Sprint 2 tuần. Dưới đây là mô phỏng của Sprint 3 - tập trung vào tính năng “Quản lý bài tập và chấm điểm”.

84.2. Sprint 3: Quản lý bài tập và chấm điểm

84.2.1. Sprint Goal

“Phát triển hoàn chỉnh chức năng quản lý bài tập cho giảng viên và chức năng nộp bài, xem điểm cho học viên”

84.2.2. Sprint Backlog

User Story	Story Points	Assignee	Status
Giảng viên tạo bài tập mới	5	Khang	Done
Giảng viên set deadline cho bài tập	3	Huy	Done
Học viên xem danh sách bài tập	3	Hiệp	Done
Học viên nộp bài (upload file)	8	Phúc	Done
Giảng viên xem bài nộp	5	Tùng	Done
Giảng viên chấm điểm và comment	8	Khang	In Progress
Học viên nhận thông báo điểm	5	Huy	To Do
API validation cho file upload	3	Backend team	Done

Tổng Story Points: 40 (Velocity dự kiến: 35-40)

84.2.3. Sprint Planning Meeting (Ngày 1 Sprint)

- Thời gian: 2 giờ
- Tham gia: Product Owner (PO), Scrum Master, Development Team
- Nội dung:
 1. PO trình bày các User Story ưu tiên cao từ Product Backlog
 2. Team ước lượng Story Points cho từng story (Planning Poker)
 3. Team commit với 40 points dựa trên velocity Sprint trước (38 points)
 4. Phân công task cụ thể cho từng thành viên
 5. Xác định Definition of Done cho Sprint
- Definition of Done:
 - Code review hoàn tất
 - Unit test coverage $\geq 70\%$
 - Integration test pass

- UI đã được PO approve
- Tài liệu API cập nhật

84.2.4. Daily Standup (Ngày 5 Sprint)

- Thời gian: 9:00 AM, 15 phút
- Minh (Frontend Dev):
 - *Yesterday*: Hoàn thành UI tạo bài tập, đã integrate với API
 - *Today*: Sẽ làm chức năng set deadline và reminder
 - *Blockers*: Không có
- Lan (Frontend Dev):
 - *Yesterday*: Implement chức năng upload file, gặp issue về file size limit
 - *Today*: Sửa bug upload, test với nhiều định dạng file
 - *Blockers*: Cần confirm từ PO về file size tối đa
- Hùng (Backend Dev):
 - *Yesterday*: API xem bài nộp đã xong, đang làm API chấm điểm
 - *Today*: Hoàn thành API chấm điểm, thêm validation
 - *Blockers*: Chờ Lan test xong file upload để verify end-to-end
- Thảo (Backend Dev):
 - *Yesterday*: Nghiên cứu về push notification
 - *Today*: Implement notification service
 - *Blockers*: Cần Firebase setup từ DevOps
- Action items:
 - PO sẽ confirm file size limit (tối đa 10MB)
 - Scrum Master liên hệ DevOps về Firebase

84.2.5. Sprint Review (Ngày 14 Sprint - cuối Sprint)

- Thời gian: 1.5 giờ
- Tham gia: Team + Stakeholders (PO, giảng viên, quản lý)
- Demo:
 1. Khang demo: Giảng viên tạo bài tập, set deadline → ✓ Approved
 2. Huy demo: Học viên xem danh sách, nộp bài → ✓ Approved
 3. Hiệp demo: Giảng viên xem bài nộp, chấm điểm → ⚠ Feedback: thêm tính năng chấm nhanh cho nhiều bài
- Kết quả:
 - Completed: 32 story points / 40 (80%)
 - Remaining: 8 points chuyển sang Sprint 4
 - PO hài lòng với tiến độ, yêu cầu minor changes

84.2.6. Sprint Retrospective (Ngày 14 Sprint - sau Review)

- Thời gian: 1 giờ
- Format: Start-Stop-Continue
- What went well (Continue):
 - Daily standup đúng giờ, hiệu quả
 - Code review giúp phát hiện bug sớm
 - Giao tiếp team tốt, hỗ trợ lẫn nhau
- What didn't go well (Stop):
 - Ước lượng story points còn lạc quan (over-commit)
 - Một số task bị block vì chờ dependency (Firebase setup)
 - Technical debt tích lũy (cần refactor module upload)
- Action items for next Sprint:
 1. Giảm velocity xuống 35 points để realistic hơn
 2. Setup Firebase sớm hơn, identify dependencies từ Sprint Planning
 3. Dành 10% thời gian Sprint cho refactoring
 4. Thêm buffer time cho integration testing

84.3. Metrics và Hiệu suất Sprint

84.3.1. Burndown Chart

Sprint 3 Burndown cho thấy:

- Ngày 1-3: Tiến độ chậm (team đang setup environment)
- Ngày 4-9: Tiến độ tốt, đúng kế hoạch
- Ngày 10-12: Xuất hiện blocker (Firebase), velocity giảm
- Ngày 13-14: Team tăng tốc, nhưng không kịp hoàn thành hết
- Bài học: Cần identify blockers sớm hơn trong Sprint Planning

84.3.2. Velocity Tracking

Sprint	Committed Points	Completed Points
Sprint 1	30	28
Sprint 2	35	38
Sprint 3	40	32
Sprint 4 (dự kiến)	35	TBD

Average Velocity: 32.7 points/sprint

84.4. Bài học từ Sprint Simulation

84.4.1. Về Planning

- Không nên over-commit dựa trên Sprint tốt nhất
- Cần xem xét dependencies giữa các task
- Buffer time cho integration và testing là cần thiết

84.4.2. Về Execution

- Daily standup giúp phát hiện blockers sớm
- Code review nên được ưu tiên cao
- Technical debt cần được address thường xuyên

84.4.3. Về Collaboration

- Giao tiếp stakeholders thường xuyên tránh hiểu lầm
- Sprint Review giúp điều chỉnh direction kịp thời
- Retrospective thực sự giúp team improve liên tục

85. Bài học

85.1. Về Quản lý Dự án

85.1.1. Lập kế hoạch chi tiết từ đầu

- Thách thức: Ban đầu nhóm ước lượng thời gian và chi phí khá lạc quan, dẫn đến phải điều chỉnh nhiều lần trong quá trình thực hiện.
- Giải pháp: Áp dụng nhiều phương pháp ước lượng (Parametric, Analogous, Three-point) để có con số realistic hơn. Thêm buffer 10-20% cho mỗi task.
- Bài học: Luôn chuẩn bị cho worst-case scenario. Sử dụng dữ liệu từ dự án cũ (nếu có) để ước lượng chính xác hơn.

85.1.2. Quản lý Stakeholder hiệu quả

- Thách thức: Giảng viên và quản lý có kỳ vọng khác nhau về tính năng, dẫn đến xung đột ưu tiên.
- Giải pháp: Tổ chức workshop để align kỳ vọng, sử dụng MoSCoW method (Must have, Should have, Could have, Won't have) để prioritize.
- Bài học: Communication là chìa khóa. Demo sớm, demo thường xuyên để tránh surprise lúc cuối.

85.1.3. Kiểm soát Scope Creep

- Thách thức: Stakeholders liên tục đề xuất tính năng mới (chatbot AI, video conferencing) ngoài phạm vi.
- Giải pháp: Xây dựng Change Request Process, đánh giá impact về thời gian, chi phí trước khi chấp nhận. Nói “không” một cách chuyên nghiệp.
- Bài học: Scope creep là kẻ thù số 1 của dự án. Cần có tài liệu rõ ràng về phạm vi và quyền quyết định.

85.2. Về Phát triển Phần mềm

85.2.1. Thiết kế Database từ đầu

- Thách thức: Thiết kế database ban đầu chưa tính đến scalability, phải refactor lại khi user tăng.
- Giải pháp: Tham khảo best practices, áp dụng normalization, thiết kế index hợp lý từ đầu.
- Bài học: Technical debt sẽ quay lại “trả thù” sau này. Đầu tư thời gian thiết kế đúng từ đầu sẽ tiết kiệm effort về lâu dài.

85.2.2. Code Review và Testing

- Thách thức: Một số bug nghiêm trọng chỉ phát hiện ở giai đoạn UAT, gây delay.
- Giải pháp: Bắt buộc code review cho mọi PR, viết unit test song song với code, integration test sau mỗi Sprint.
- Bài học: Quality không thể được “test vào” sản phẩm. Phải build quality từ đầu (shift-left testing).

85.2.3. Documentation

- Thách thức: Tài liệu API không được cập nhật kịp thời, gây khó khăn cho frontend dev.
- Giải pháp: Sử dụng tools tự động (Swagger/OpenAPI), coi documentation là part of Definition of Done.
- Bài học: Documentation không phải optional. Nó giúp onboarding member mới và maintain dự án lâu dài.

85.3. Về Teamwork

85.3.1. Giao tiếp trong Team

- Thách thức: Hiểu lầm giữa frontend và backend về API contract, dẫn đến rework.
- Giải pháp: Daily standup, shared documentation, pair programming cho phần phức tạp.
- Bài học: Over-communicate chứ đừng under-communicate. Giả định là nguyên nhân của mọi confusion.

85.3.2. Phân công công việc

- Thách thức: Một số member bị overload trong khi người khác rảnh, gây mất cân bằng.
- Giải pháp: Sprint Planning rõ ràng, theo dõi capacity của từng người, ready to adjust mid-Sprint.
- Bài học: Team velocity không phải tổng của individual capacity. Cần tính đến collaboration overhead.

85.3.3. Continuous Improvement

- Thách thức: Team có xu hướng repeat lỗi cũ do không có cơ chế học hỏi.
- Giải pháp: Sprint Retrospective nghiêm túc, track action items, review action items ở Sprint sau.
- Bài học: Retrospective không phải formality. Nếu không có action items cụ thể thì meeting đó vô nghĩa.

85.4. Về Công nghệ

85.4.1. Lựa chọn Tech Stack

- Thách thức: Chọn framework mới nhất nhưng thiếu community support, gặp nhiều bug.
- Giải pháp: Balance giữa modern và stable. Chọn tech có community lớn, tài liệu đầy đủ.
- Bài học: “Boring technology” thường là lựa chọn an toàn cho production. Experimental tech nên dùng cho side project.

85.4.2. Cloud và Deployment

- Thách thức: Chi phí cloud tăng đột biến do không optimize resource.
- Giải pháp: Monitor resource usage, áp dụng auto-scaling, sử dụng spot instance cho non-critical workload.
- Bài học: Cloud is not free. Cần cost monitoring và optimization ngay từ đầu.

85.5. Khuyến nghị cho Dự án Tương lai

1. • Đầu tư vào Planning: 20% thời gian dự án nên dành cho planning và thiết kế
2. • Embrace Agile mindset: Linh hoạt với thay đổi nhưng không đồng nghĩa với không có kế hoạch
3. • Build MVP first: Tập trung vào core features, tránh feature bloat
4. • Automate everything: CI/CD, testing, deployment để giảm manual effort
5. • Invest in people: Training, knowledge sharing giúp team improve liên tục
6. • Risk management: Identify risks sớm, có mitigation plan
7. • Customer-centric: Build what users need, not what we think they need

85.6. Kết luận

Dự án “Ứng dụng quản lý lớp học trực tuyến” là một trải nghiệm quý giá về quản lý dự án phần mềm. Chúng ta đã học được:

- Tầm quan trọng của planning và documentation
- Lợi ích của Agile trong môi trường thay đổi
- Vai trò then chốt của communication và stakeholder management
- Giá trị của continuous improvement qua Retrospective
- Cân bằng giữa quality và deadline

Những bài học này không chỉ áp dụng cho dự án này mà còn là hành trang cho các dự án trong tương lai. Success của dự án không chỉ đo bằng sản phẩm delivered mà còn ở process improvement và team growth.

86. Tài liệu tham khảo

1. Project Management Institute (PMI). *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 7th Edition.
2. Ken Schwaber & Jeff Sutherland. *The Scrum Guide - The Definitive Guide to Scrum: The Rules of the Game*, 2020.
3. Google Classroom Documentation. <https://edu.google.com/products/classroom/>
4. Moodle Official Documentation. <https://docs.moodle.org/>
5. Martin Fowler. *Refactoring: Improving the Design of Existing Code*, 2nd Edition, Addison-Wesley, 2018.
6. Eric Ries. *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, 2011.

87. Retrospective C

87.1. Đánh giá trải nghiệm Agile

- Lần đầu trải nghiệm Scrum với phân vai rõ ràng: PO, SM, Dev Team.
- Daily Scrum giúp nắm bắt tiến độ và blockers nhanh hơn Waterfall.
- Sprint ngắn (1 tuần) tạo áp lực nhưng cũng tăng sự tập trung.
- Việc review và retrospective sau mỗi Sprint giúp cải thiện liên tục.

87.2. Những khác biệt so với Waterfall

- Waterfall: kế hoạch cố định, thay đổi khó; Agile: linh hoạt, thích ứng nhanh.
- Waterfall: feedback cuối dự án; Agile: feedback sau mỗi Sprint.
- Waterfall: tài liệu nặng; Agile: ưu tiên sản phẩm hoạt động.
- Agile phù hợp hơn với dự án có requirement thay đổi thường xuyên.

87.3. Những việc làm tốt

- Phân vai Scrum rõ ràng: PO (Phúc), SM (Khang, Hiệp), Dev Team (Huy, Tùng).
- Tổ chức 2 buổi Daily Scrum (Sprint 2, 3) với biên bản đầy đủ.
- Khang hoàn thành nhiệm vụ sớm nhất nhóm (2025/11/12).
- Xác định blockers và action items cụ thể sau mỗi Daily Scrum.
- Team phối hợp tốt giữa FE & BE trong phát triển tính năng.

87.4. Những việc làm chưa tốt

- API contract chưa thống nhất giữa FE & BE, gây trễ integration.
- Naming convention chưa đồng bộ giữa các module.
- Một số thành viên bận lịch học, ảnh hưởng timeline Sprint.
- Thiếu môi trường staging cho integration tests.
- Giao tiếp ngoài Daily Scrum còn hạn chế.

87.5. Đánh giá đóng góp từng thành viên

87.5.1. Hiệp

- Vai trò: Scrum Master
- Theo dõi tiến độ integration và DevOps.
- Follow-up Firebase notification.

87.5.2. Khang

- Vai trò: Scrum Master
- Hoàn thành nhiệm vụ của bản thân sớm nhất nhóm
 - Nhóm trưởng chia việc xong vào lúc 2025-11-12 20:07:00
 - Hoàn thành vào lúc 2025-11-12 22:45:00
 - Tổng thời gian: 2.63 tiếng
- Hỗ trợ kiểm tra CI pipeline và code review.

87.5.3. Huy

- Vai trò: Developer (Backend)
- Hoàn thiện API CRUD, chấm điểm, xử lý conflict detection.
- Viết unit test và deploy staging.

87.5.4. Phúc

- Vai trò: Product Owner
- Xác nhận requirements, acceptance criteria.
- Chuẩn bị UAT checklist và test cases.

87.5.5. Tùng

- Vai trò: Developer (Frontend)
- Thiết kế UI flow và tích hợp frontend.
- Triển khai notification UI và test upload.

87.6. Bài học rút ra

- Cần thống nhất API contract và naming convention trước khi code.
- Môi trường staging cần thiết lập sớm.
- Scrum hiệu quả khi team giao tiếp thường xuyên, không chỉ trong Daily Scrum.
- Timeline Sprint cần linh hoạt theo lịch học của thành viên.

87.7. Đề xuất cải thiện

- Tạo API contract document trước mỗi Sprint.
- Thiết lập môi trường staging ngay từ đầu.
- Tăng cường giao tiếp nhóm qua kênh chat riêng.
- Ghi nhận feedback để cải thiện Sprint tiếp theo.
- Cập nhật timeline linh hoạt theo lịch học của thành viên.