**Group Project Reflection #2**

**PART A: TEAM REFLECTION (Equivalent to 800 words/team)**

***Reflect on the below question as a team.***

|  |  |  |
| --- | --- | --- |
|  |  |  |
| ERROR-404 | **Team member name** | **Student ID** |
| Team member 1 | CALE FRASER | 21605948 |
| Team member 2 | ABDUL RAHMAN AHAMED LUKMAN | 22020948 |
| Team member 3 | ANKAN DAS | 22018897 |
| Team member 4 | MD ARAFATH REJA RIFAT | 22035298 |
| Team member 5 | PRACHAURJA SARKER | 21781846 |

1. **Sprint 2 Task List**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Task** | **Assignee** | **Status** |
| 1 | Success Factors and Criteria | ABDUL RAHMAN AHAMED LUKMAN | **Completed** |
| 2 | Resource Allocation | MD ARAFATH REJA RIFAT | **Completed** |
| 3 | Risk Management | CALE FRASER | **Completed** |
| 4 | Communication Plan | PRACHAURJA SARKER | **Completed** |
| 5 | Stakeholder Identification | ANKAN DAS | **Completed** |

1. **What worked well? What didn’t go so well**

Error-404 communicated effectively via WhatsApp chats and meetings, with thorough communication before selecting Project 2 (PhishShield). All members were involved and completed their dedicated tasks, assigning based on each members preference.

Error-404 hit a wall with scheduling. We pushed a late-night meeting to “get it done,” but shift work and other unit deadlines meant several members couldn’t attend, updates went stale, and a couple of tasks slipped or were duplicated. The root cause wasn’t commitment, it was capacity: jobs + assignments + mismatched time zones within Melbourne routines made our cadence unrealistic. We reset the system mid-sprint: two fixed meeting slots that fit most rosters, a hard “no meetings after 10 pm” rule, and a daily async check-in on WhatsApp (Yesterday/Today/Blocked by 10 am). Confluence became the decision log; Jira capacity flags were added to each ticket; every critical item now has a backup owner; and we set an internal deadline 24 hours before the real one to avoid last-minute chaos.

1. **Review and Reason**

**Problem:** Our scheduling failed. We set a late-night meeting to “push through,” but shift work and overlapping assignment deadlines meant several members couldn’t attend. Outcomes: stale updates, duplicated effort, and a couple of late/partial submissions.

**Why it happened (root causes):**

* **Fantasy capacity planning:** We scheduled to ideal availability, not real hours after jobs and assignments.
* **No resource calendar:** We didn’t capture each person’s constraints (work shifts, exam weeks), so clashes were predictable but unseen.
* **Over-reliance on synchronous meetings:** Critical decisions depended on everyone being live; when people missed, context got lost.
* **Weak ownership signals:** Tasks lacked clear primary + backup owners, so others “filled the gap,” creating overlap.
* **No buffer:** We aimed for the external deadline with zero internal slack, so any slip became a scramble.

**How this maps to Project Management concepts:**

**Resource & capacity planning / resource levelling:** We skipped levelling people’s workloads across time, so the “plan” ignored real capacity. Textbook miss.

**Schedule management (buffers & critical path thinking):** No internal buffer before the due date; we treated every task as if it had float. It didn’t.

**Risk management:** “Student jobs + assessment peaks” is a high-likelihood risk. We didn’t log it, assign an owner, or pre-define mitigations.

**Communications management plan:** We relied on ad-hoc WhatsApp + meetings without a defined async protocol or a single source of truth, so version drift and duplicate work were inevitable.

**Roles & responsibility (RACI):** Ambiguity around who owns what (and who backs them up) led to shadow ownership and rework.

**Agile practice (sprint capacity, WIP limits, DoD):** We planned work by story count, not by available hours; no WIP limits; “Done” didn’t include “communicated + documented,” so hand-offs broke.

1. **Building on the lessons from Sprint-1**

***Lesson 1 - More meetings ≠ better coordination.***In Sprint 1 we crammed in late-night meetings; attendance dropped (jobs + other units), context went stale, and drafts needed heavy re-edits.   
**Change (Sprint 2):** Time-boxed meetings at reasonable hours, fixed agenda, hard stop; daily async update (Yesterday/Today/Blocked by 10am) so absent members still move work. Decisions captured in Confluence.  
**PM link:** Communications Management Plan; Agile ceremony hygiene.  
**Impact:** Fewer no-shows, faster decisions, far less duplicated work.

***Lesson 2 - Tooling friction was the bottleneck (Confluence/Jira).***Some members were still unsure how to structure pages or tickets; this created rework and “please fix format” loops.  
**Change (Sprint 2):** 15-minute micro-clinics + pair-ups; created templates (Confluence meeting note/decision log, Jira ticket with Definition of Ready/Done + acceptance criteria).  
**PM link:** Knowledge management; process standardisation.  
**Impact:** Consistent pages, clearer tickets, fewer editing passes to “make it publishable.”

***Lesson 3 - Ambiguous ownership caused overlap.***In Sprint 1, tasks bounced around; people “helped” by re-doing work already assigned.  
**Change (Sprint 2):** RACI on each deliverable (Owner, Backup, Reviewer), visible on the Confluence page header; Jira enforces primary + backup assignee.  
**PM link:** Roles & Responsibilities (RACI).  
**Impact:** Clean hand-offs; zero double-work on core tasks.

***Lesson 4 - We planned for ideal capacity, not real life.***Jobs and other assignments collided with our sprint scope.  
**Change (Sprint 2):** Simple availability calendar; sprint scope sized to actual hours; WIP limits; internal “T-24h” deadline before the real one.  
**PM link:** Resource/capacity planning, buffer management.  
**Impact:** Fewer last-minute scrambles; on-time submissions improved.

***Lesson 5 - Heavy edits were a symptom, not the disease.***Drafts needed “more editing for each” because expectations weren’t explicit.  
**Change (Sprint 2):** Lightweight style guide + QA checklist (structure, citations, figures, traceability to requirements); two-step review (peer, then lead).  
**PM link:** Quality management; Definition of Done.  
**Impact:** Shorter review cycles; higher first-pass acceptance.

1. **Reflecting on the use of sprints to complete work**

**What worked**

* Timeboxes forced focus and gave us a shared rhythm.
* Definition of Done + peer review cut sloppy hand-offs.
* Regular checkpoints made blockers visible earlier.

**What didn’t**

* Student reality (jobs + other units) blew up attendance → sprint ceremonies became overhead.
* We planned to “ideal” hours; rollover stories piled up.
* Late stakeholder changes clashed with fixed sprint scopes (scope creep mid-sprint).
* Tooling learning curve (Jira/Confluence) slowed velocity; “more meetings” didn’t fix it.

**Net effectiveness**

* After we tightened comms (async Y/T/B check-ins), added a T-24h internal buffer, and used RACI + backup owners, duplication dropped, and we hit the next milestone on time. Still, pure Scrum cadence was fragile under variable capacity.

# **What we’d use next time**

**Pick:** Scrumban with stage-gate checkpoints (not pure Scrum, not pure Kanban, not Waterfall).

**Why this fits our constraints**

* **Capacity swings:** Kanban-style WIP limits (e.g., max 2 tickets/person) respect real availability; no ceremony bloat.
* **Hard academic deadlines:** Light **stage gates** (Design freeze → Build freeze → QA freeze) keep us deadline-driven without big-bang Waterfall.
* **Scope churn:** Pull-based flow absorbs late feedback better than locking scope for a full sprint.
* **Quality + traceability:** Retain Scrum’s DoD, templates, and Peer → Lead two-step review so deliverables are publishable.

# 

# **How we’d run it**

* **Cadence:** 1-week flow cycles, not heavy 2-week sprints.  
  + Mon 20–30 min planning (pull to WIP, capacity-based).
  + Daily async Y/T/B by 10am; no meetings after 10pm.
  + Wed mini-refinement (15 min); Fri gate check (move items to next stage or unblock).
* **Board & policies:** Backlog → Ready → In Progress (WIP 2/person) → Review → Done. Blocked gets a flag + owner within 24h.
* **Gates aligned to unit deliverables:**
  + **G1 Design freeze** (content agreed, figures chosen).
  + **G2 Build freeze** (writing/implementation done, only QA/edits allowed).
  + **G3 QA freeze** (citations, formatting, accessibility checks complete).
* **Risk & buffers:** Keep a tiny risk log (top 5 with owners); internal T-24h buffer before any submission.
* **Roles:** RACI on each page/ticket (Owner, Backup, Reviewer). No ticket without a backup.
* **Metrics that matter:** Cycle time (Start → Done), WIP aging, and % carried-over items. If aging > 3 days, we swarm or descope.

**PART B: INDIVIDUAL REFLECTION (Equivalent to 150 words/student)**

The below section must be filled out **individually by each team member.**

Reflect on your performance during the final sprint by considering the following: your personal contributions and the quality and impact of your work; how well you integrated with and supported your team; the effectiveness of your communication, including giving and receiving feedback; your problem-solving and innovative solutions; your adaptability to changes; your conflict resolution skills and their outcomes; how your efforts aligned with and supported the team and project goals; your personal growth, including new skills acquired and areas for future improvement; and the overall lessons learned and key takeaways from the project experience.

|  |
| --- |
| **Team member 1:** |
| **Team member 2:** |
| **Team member 3:** |
| **Team member 4: MD ARAFATH REJA RIFAT** |
| **Team member 5:** |