

AKG SW DEPARTMENT



# Agile Workflow Overview

Moving Forward



## + Objective

- To Build a unified, agile software team that delivers high-quality, innovative solutions with speed and reliability.
- Together, we aim to drive excellence, ownership, and continuous improvement in every project we deliver.



# Agile Workflow Overview

+ (SCRUM + KANBAN HYBRID)

This model combines **Scrum's structure** (sprints, planning, review) with **Kanban's flexibility** (continuous flow, WIP limits).

The flow ensures:

- Clear ownership.
- Fast feedback cycles.
- Traceability and visibility.



# Roles & Responsibilities

## Workflow

Role	Core Responsibility	Key Deliverables
Software Manager	Strategic direction, project oversight, and prioritization across teams	Sprint goals, resource allocation, project roadmap
System Analyst	Convert business needs into system requirements and user stories	BRD, system specs, use cases, user stories
Technical Team Leader	Translate requirements into technical tasks, assign and guide team	Sprint backlog, technical designs, code reviews
Software Team	Execute development tasks, test, and document	Features, bug fixes, unit tests, commits, documentation



# Step-by-Step

## Agile Flow

### + Requirement Intake & Analysis ①

**Owner:** System Analyst

**Flow:**

- Receive business requirement or feature request.
- Discuss scope with Software Manager.
- Create:
  - Business Requirement Document (BRD) (**what the business wants**)
  - Functional Specification / User Stories (**what the system should do**)
  - Output: User stories in backlog.
- Review & approve by Software Manager.

+

**Output:** User stories in backlog. approved by Software Manager.



# Step-by-Step

## Agile Flow

### + Technical Planning & Sprint Setup ②

**Owner:** Technical Team Leader

**Flow:**

- Analyze user stories with team.
- Define technical approach & dependencies.
- Break down into development tasks (e.g., database, backend, UI).
- Hold Sprint Planning Meeting:
  - Confirm sprint goals.
  - Assign tasks.
  - Set sprint duration (typically 2 weeks).

**Output:** Sprint backlog approved by Software Manager.



# Step-by-Step

## Agile Flow

### + Development Execution ③

**Owner:** Technical Team Leader + Software Team

**Flow:**

- Daily stand-up (15 min **max**):
  - What I did yesterday
  - What I'll do today
  - Any blockers
- Technical Team Leader tracks progress and removes obstacles.
- System Analyst provides clarifications when needed.

**Backlog → In Progress → Code Review → Testing → Done**



# Step-by-Step

## Agile Flow

### + Testing & Quality Review ④

**Owner:** Technical Team Leader + Tester Engineer

**Flow:**

- Developer completes task > submits for Code Review.
- Q.Engineer (manual or automated).
- Bugs go back to backlog (prioritized by Team Lead).
- System Analyst validates functionality vs requirements.

**Output:** Verified feature ready for release.



# Step-by-Step

## Agile Flow

### + Sprint Review & Demo ⑤

**Owner:** Software Manager + System Analyst

**Flow:**

- End of sprint: team(\*) presents completed features.
- Stakeholders give feedback.
- System Analyst notes enhancement ideas.
- Software Manager updates roadmap and priorities.

**Output:** Approved deliverables + input for next sprint..



# Step-by-Step

## Agile Flow

### + Sprint Retrospective ⑥

**Owner:** Technical Team Leader

**Flow:**

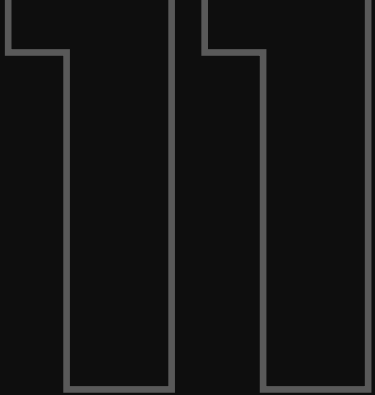
- Team discusses:
  - What went well.
  - What didn't.
  - What can be improved next sprint.
- Capture 2–3 improvement actions.

**Output:** Continuous process improvement.



# Weekly Flow

## (2-week Sprint)



Day	Key Focus
Sunday	Sprint planning & backlog grooming
Mon-Wed	Development + Daily stand-ups
Thursady	Mid-sprint sync & code reviews
Next Sunday	Final testing & sprint review
Next Monday	Retrospective + new sprint setup



# Role Descriptions & Reporting Lines

## + Software Manager ①

**Reports to:** CTO / ICT Director

**Manages:** Technical Team Leader and System Analyst

**Responsibilities:**

- Define the overall *software roadmap* and delivery strategy.
- Manage resources, budgets, and timelines.
- Approve requirements, sprint scopes, and priorities.
- Ensure integration between software projects, data, and company systems.

**Decision Level:** Strategic + Tactical



## Role Descriptions & Reporting Lines

### Technical Team Leader 2

**Reports to:** Software Manager

**Manages:** Software Development Team

**Coordinates with:** System Analyst

#### Responsibilities:

- Translate requirements into technical tasks and architecture.
- Manage daily standups, sprint progress, and task assignments.
- Ensure coding standards, code reviews, and performance optimization.
- Resolve blockers and align technical solutions with business goals.
- Mentor developers and ensure delivery quality.
- Handling all team daily paperwork & documentation.

**Decision Level:** Tactical + Operational



# Role Descriptions & Reporting Lines

## + System Analyst ③

**Reports to:** Software Manager (functionally)

**Work closely with:** Technical Team Leader and business stakeholders

### **Responsibilities:**

- Gather and analyze business requirements.
- Convert requirements into user stories and technical specs.
- Validate sprint outcomes against business goals.
- Support testing and documentation for user acceptance.

**Decision Level:** Functional / Analysis



# Role Descriptions & Reporting Lines

## + Software Development Team ④

**Reports to:** Technical Team Leader

**Work closely with:** Backend, Frontend, Database, QA, UI/UX, etc.

**Responsibilities:**

- Develop, test, and deliver assigned sprint tasks.
- Participate in daily standups and sprint reviews.
- Follow coding standards and documentation practices.
- Collaborate closely with analyst and lead for feature completion.

**Decision Level:** Operational



# Reporting line

## Summary

+

## Reporting Line Summary

Role	Reports To	Collaborates With	Key Focus
Software Manager	CTO / ICT Director	All roles	Strategy, prioritization, roadmap
Technical Team Leader	Software Manager	System Analyst, Dev Team	Technical planning & execution
System Analyst	Software Manager	Team Leader, Users	Requirement translation & validation
Software Developers	Technical Team Leader	QA, Analyst	Coding, testing, delivery



# Communication Plan



Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Microsoft Teams (Channels & Meetings) + Emails	<ul style="list-style-type: none"><li>- Daily communication on tasks and project updates</li><li>- Sprint planning and reviews</li><li>- Quick alignment or clarifications</li></ul>	All Team Members	Daily / As needed	<div><div></div> Normal</div>	<ul style="list-style-type: none"><li>- Use dedicated channels per project/sprint</li><li>- Keep updates concise</li><li>- Record meeting notes in channel posts</li></ul>



# Communication Plan

18

Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Microsoft Teams (Calls / Video Meetings)	<ul style="list-style-type: none"><li>- Standups, retrospectives, or ad-hoc sync meetings</li><li>- Troubleshooting sessions</li></ul>	Team Leader, Developers , Analyst	Daily / As needed	● Normal	<ul style="list-style-type: none"><li>- Keep meetings within 15–30 minutes</li><li>- Record if needed for absent members</li></ul>



# Communication Plan

Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Email (Official Communication)	<ul style="list-style-type: none"><li>- Formal updates and reports</li><li>- Documented decisions, approvals, and escalation</li><li>- Communication with external departments or management</li></ul>	Software Manager, Team Leader, Analyst	Weekly / As needed	● Standard / Formal	<ul style="list-style-type: none"><li>- Use clear subject lines and summary format</li><li>- CC only relevant people</li><li>- Archive and track decisions</li></ul>



# Communication Plan

20



Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Email (Incident & Issue Escalation)	<ul style="list-style-type: none"><li>- Reporting major bugs, production issues, or delays</li><li>- Escalating blocked or critical dependencies</li></ul>	Team Leader > Manager	As needed	● High	<ul style="list-style-type: none"><li>- Include full description, screenshots, and ticket link</li><li>- Mark as "High Importance"</li></ul>



# Communication Plan

20

Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
WhatsApp (Urgent Only)	<ul style="list-style-type: none"><li>- Urgent or time-sensitive issues during/after working hours</li><li>- Server down, release failure, or urgent approval</li></ul>	Team Leader, Manager, Key Devs	As needed	● Critical	<ul style="list-style-type: none"><li>- Keep messages concise and factual</li><li>- Follow up with Teams/email summary after resolution</li><li>- Avoid non-work use</li></ul>



# Communication Plan

20



Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Shared Drives / OneDrive	- Document storage and version control	Analyst, Team Leader, Developers	Continuous	● Standard	- Store design docs, meeting minutes, and technical specs - Maintain folder discipline



# Communication Plan

Channel / Tool	Purpose	Used By	Frequency	Urgency Level	Guidelines & Notes
Weekly Sync Meeting	<ul style="list-style-type: none"><li>- Review sprint progress and blockers</li><li>- Align next sprint priorities</li></ul>	Manager, Analyst, Team Leader	Weekly	<div><div></div> Standard</div>	<ul style="list-style-type: none"><li>- Share agenda 1 day before</li><li>- Record decisions and assign owners</li></ul>



# Communication Plan Summary



- Use **Teams** for day-to-day collaboration, not for official approvals.
- Use **Email** for decisions, approvals, and formal escalations.
- Use **WhatsApp** only for urgent issues that need immediate action.
- Always follow up urgent chats with a **formal record** (Teams or Email).
- Keep **response time standards** (e.g., Teams within 2 hours, email within 24 hours, WhatsApp within 30 minutes).





# Let's Work Together

