Batch: D-2 **Roll No.:** 1601012151

Experiment / assignment / tutorial No 8

TITLE: Project Plan document for Mini Project

AIM: To learn and understand the way of developing the software by classical methods of software engg., Planning and monitoring of the project using tools and prepare a document for the same by using the concept of software engineering

Expected OUTCOME of Experiment:

Books/ Journals/ Websites referred:

1. Roger Pressman, Software Engineering: A practitioners Approach, McGraq Hill, 2010 ,6th edition
2. Ian Somerville , Software Engineering , Addison Wesley,2011,9th edition
3 http://en.wikipedia.org/wiki/Software_requirements_specification

Software Project Management Plan

for

<HealthCare Management System>

<Nikhil Patil & Hyder Presswala>

<08-11-2024>

Version	Release Date	Responsible Party	Major Changes
0.1	*08-11-2024*	KJSCE	Initial Document Release for Comment

Table of Contents

1. Introduction

Software Project Management Plan (SPMP) provides an overview of the **Healthcare Management System (HMS)** project, detailing the project's objectives, deliverables, and the process for managing and evolving the project plan.

1.1 Project Overview

The **Healthcare Management System (HMS)** is designed to improve the efficiency of healthcare operations by providing a comprehensive platform for managing patient data, appointments, billing, and communication between healthcare professionals. The primary objectives of the project are to streamline administrative tasks, improve patient care coordination, and ensure compliance with healthcare data privacy regulations.

- **Major Work Activities**:
  - **Requirement Analysis**: Gather detailed requirements from stakeholders.
  - **System Design**: Design system architecture, user interface, and database structure.
  - **Development**: Develop core features, including patient management, appointment scheduling, and billing.
  - **Testing and Deployment**: Conduct testing to ensure system functionality, followed by deployment.
- **Major Milestones**:
  - **Requirement Gathering**: Month 1
  - **System Design Completed**: Month 2
  - **Prototype Completion**: Month 3
  - **Beta Testing**: Month 5
  - **Final System Deployment**: Month 6
- **Required Resources**:
  - **Human Resources**: Project manager, developers, healthcare domain experts, testers, system administrators.
  - **Technological Resources**: Web development tools, database management system (DBMS), security infrastructure.
- **Budget**: The estimated project budget is **$50,000**, covering software development, testing, and deployment costs.

The HMS project is **independent** but will integrate with existing **electronic health records (EHR)** systems and hospital management platforms in the future.

1.2 Project Deliverables

The primary deliverables for the **Healthcare Management System (HMS)** project are:

- **System Requirements Document** – Completion by **Month 1**.
- **System Design Document** – Completion by **Month 2**.
- **Working Prototype** – Delivery by **Month 3**.
- **Final System Deployment** – Delivery by **Month 6**.

These deliverables will be provided to the project stakeholders, including healthcare providers and administrators, via **email and cloud storage**.

1.3 Evolution of the SPMP

The SPMP will evolve throughout the project's lifecycle. The initial version will be finalized at the beginning of the project and disseminated to all team members and stakeholders for review.

- **Updates**:
  - Scheduled updates will be made at the end of each development phase to incorporate feedback and track progress.
  - Unscheduled updates will be applied as needed based on changes in scope, resource availability, or unforeseen risks. All changes to the SPMP will be logged and reviewed by the project manager and key stakeholders.

1.4 Reference Materials

- **Project Requirements Document** – Document detailing the functional and non-functional requirements for the HMS system. (Date: Month 1, Author: Project Manager)
- **HIPAA Compliance Guidelines** – Federal guidelines for data security and privacy in healthcare. (Date: Ongoing, Author: U.S. Department of Health & Human Services)
- **GDPR Data Protection Regulations** – European Union regulations for data protection. (Date: Ongoing, Author: European Commission)

1.5 Definitions and Acronyms

- **HMS**: Healthcare Management System
- **EHR**: Electronic Health Records
- **HIPAA**: Health Insurance Portability and Accountability Act
- **GDPR**: General Data Protection Regulation
- **DBMS**: Database Management System

2. Project Organization

This section specifies the process model for the **Healthcare Management System (HMS)** project and describes the organizational structure that supports the project's execution.

2.1 Process Model

For the **Healthcare Management System (HMS)** project, we will follow an **Agile** process model, particularly using an **iterative and incremental development approach**. The Agile model is chosen to allow flexibility and continuous improvement throughout the project lifecycle, which is important for adapting to changes in healthcare regulations and technology.

- **Roles**:
  - **Project Manager**: Oversees the entire project, managing scope, timeline, resources, and risks.
  - **Business Analysts**: Work with stakeholders to gather and define system requirements.
  - **Software Developers**: Implement system features and perform unit testing.
  - **Quality Assurance (QA) Engineers**: Conduct system testing and ensure quality standards.
  - **System Administrators**: Responsible for system deployment, security, and maintaining the infrastructure.
  - **Healthcare Experts**: Provide domain knowledge, ensuring that the system meets healthcare industry requirements.
- **Activities**:
  - **Project Initiation**: Requirement gathering, system design, and planning.
  - **Product Development**: Iterative development cycles, including coding, testing, and reviews.
  - **Product Release**: Final testing, system deployment, and delivery to stakeholders.
  - **Project Termination**: Post-deployment support, documentation, and final project closure.
- **Entry Criteria**:
  - For **Initiation**: Signed project charter, initial requirements gathered.
  - For **Development**: Design documents approved, resources allocated.
  - For **Release**: Successful completion of acceptance testing, approval from stakeholders.
  - For **Termination**: Project objectives met, deliverables handed over, post-deployment support in place.
- **Exit Criteria**:
  - For **Initiation**: All stakeholders have agreed to project scope and timelines.
  - For **Development**: All user stories or features completed, and system meets functional requirements.
  - For **Release**: System is stable, tested, and deployed to the production environment.
  - For **Termination**: All project documentation completed, knowledge transferred, and final reports submitted.

2.2 Organizational Structure

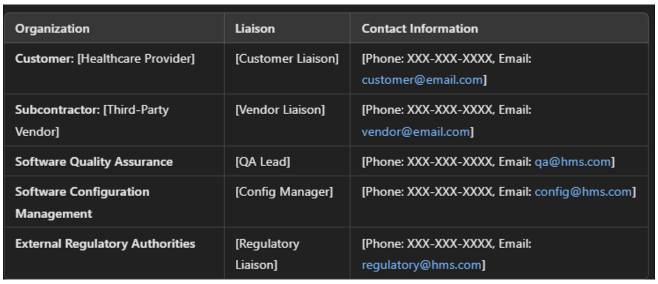
The **HMS** project will be managed by a **project management team**, with each member assigned specific responsibilities. Below is a high-level chart outlining the internal structure and authority hierarchy for the project:

```
Project Sponsor
    |
Project Manager
    |
  ├── Business Analysts
  ├── Software Developers
  ├── Quality Assurance Engineers
  ├── System Administrators
  └── Healthcare Experts
```

- **Project Sponsor**: Provides financial and strategic support for the project.
- **Project Manager**: Manages project execution, scope, resources, and risk management.
- **Business Analysts**: Liaison between stakeholders and development teams, gathering and refining requirements.
- **Software Developers**: Develop the HMS system, implement new features, and maintain code quality.
- **Quality Assurance Engineers**: Responsible for testing the system to ensure functionality and security.
- **System Administrators**: Deploy, monitor, and maintain the system in a secure and stable environment.
- **Healthcare Experts**: Ensure that the system complies with healthcare regulations and meets industry-specific needs.

2.3 Organizational Interfaces

Describe the administrative and managerial interfaces between the project and the primary entities with which it interacts. A table may be a useful way to represent this.



Organization	Liaison	Contact Information
Customer: [Healthcare Provider]	[Customer Liaison]	[Phone: XXX-XXX-XXXX, Email: customer@email.com]
Subcontractor: [Third-Party Vendor]	[Vendor Liaison]	[Phone: XXX-XXX-XXXX, Email: vendor@email.com]
Software Quality Assurance	[QA Lead]	[Phone: XXX-XXX-XXXX, Email: qa@hms.com]
Software Configuration Management	[Config Manager]	[Phone: XXX-XXX-XXXX, Email: config@hms.com]
External Regulatory Authorities	[Regulatory Liaison]	[Phone: XXX-XXX-XXXX, Email: regulatory@hms.com]

Table F-1. Project Interfaces

- **Customer**: The healthcare provider who will use the HMS system.
- **Subcontractors**: Any third-party vendors involved in providing additional

components or services for the system.

- **Software Quality Assurance**: Internal team responsible for ensuring that the software meets the required standards.
- **Software Configuration Management**: Ensures that version control, deployment, and system configuration are correctly managed throughout the project.
- **Regulatory Authorities**: Government or industry bodies that oversee the healthcare industry, ensuring compliance with regulations like **HIPAA**.

2.4 Project Responsibilities

Identify and state the nature of each major project function and activity, and identify the individuals who are responsible for those functions and activities. Tables of functions and activities may be used to depict project responsibilities.

Role	**Description**	**Person**
Project Manager	Leads Project Team; Responsible for project deliverables	Hyder Presswala
Technical Team Leader(s)	Makes sure all members are working	Nikhil Patil

Table F-2. Project Responsibilities.

3. Managerial Process

This section outlines the **management objectives**, priorities, and the approach for managing the **Healthcare Management System (HMS)** project. It will also describe the key management dimensions—**cost**, **schedule**, and **scope**—and how they will be managed throughout the project lifecycle.

3.1 Management Objectives and Priorities

The key management objectives for the **HMS** project are:

1. **Delivering a high-quality system**: The HMS system must meet the requirements of healthcare providers, ensuring it is reliable, secure, and compliant with healthcare regulations such as **HIPAA** and **GDPR**.
2. **Meeting project timelines**: Ensuring that all deliverables are completed on time, from initial requirements gathering to final system deployment. Timely delivery is essential for meeting healthcare operational deadlines.
3. **Adhering to budget constraints**: The project must be completed within the allocated budget of **$50,000**, covering development, testing, deployment, and

other associated costs.

4. **Ensuring stakeholder satisfaction**: The system should meet or exceed the expectations of stakeholders, including healthcare providers, administrators, and end-users.
5. **Managing risks effectively**: Identifying potential risks early on (e.g., data security, regulatory changes) and having mitigation strategies in place to prevent or minimize disruptions.

To ensure the project's success, we adopt the **Triple Constraint** model (also known as the **Iron Triangle** of project management), which focuses on balancing the **scope**, **schedule**, and **cost** of the project. We prioritize meeting the schedule and scope while managing costs carefully.

Project Dimension	**Fixed**	**Constrained**	**Flexible**
Cost		X	
Schedule	X		
Scope (functionality)			X

Table F-3: Flexibility Matrix

- **Cost**: The project budget is **constrained**. While cost management is essential, some flexibility may exist to accommodate unexpected issues, but the overall budget should not exceed the set limit.
- **Schedule**: The **schedule is fixed**, as meeting deadlines for system deployment is crucial for stakeholders. Delays may impact healthcare operations, so the timeline must be adhered to as closely as possible.
- **Scope (functionality)**: The **scope is flexible**. While core functionality is defined, the system may be adjusted to incorporate new features or enhancements based on feedback from healthcare professionals or changes in regulatory requirements. However, such changes must be managed carefully to avoid scope creep.

3.2 Assumptions, Dependencies, and Constraints

Assumptions

- **Stakeholder Availability**: Key stakeholders will be available for timely feedback and reviews.
- **Regulatory Stability**: No significant changes in healthcare regulations (e.g., **HIPAA**, **GDPR**) during the project.

- **Existing Systems Compatibility**: Assumes integration with current healthcare management systems will be possible.
- **Technology Availability**: Required technologies (software, cloud, hardware) will be accessible and reliable.
- **User Proficiency**: Healthcare staff will have basic computer skills and will undergo training.

Dependencies

- **Third-Party Services**: Dependent on third-party vendors for cloud hosting, authentication, and integrations.
- **Regulatory Approvals**: Must obtain necessary regulatory approvals for healthcare compliance.
- **Technology Stack**: Relies on specific tech stack availability (programming languages, cloud platforms).
- **Healthcare Provider Collaboration**: Successful collaboration from healthcare providers for requirements gathering and testing.

Constraints

- **Budget**: Fixed budget of **$50,000** for development, testing, and deployment.
- **Schedule**: Must be completed within **6 months**.
- **Scope**: Limited to core features such as **patient management**, **appointment scheduling**, and **billing**; additional features can be added in future phases.
- **Regulatory Compliance**: Must comply with healthcare regulations like **HIPAA** and **GDPR**.
- **System Performance**: Must support a specified volume of users without performance issues.

Priorities: Functionality, Schedule, Budget

- **Fixed**: **Schedule** (on-time delivery is critical).
- **Constrained**: **Budget** (must stay within the $50,000 limit).
- **Flexible**: **Scope** (additional features can be considered later

3.3 Risk Management

Risk management is a critical process for identifying, analyzing, and mitigating risks that may impact the successful delivery of the **Healthcare Management System (HMS)** project. The following steps will be used to manage risks:

Risk Identification

- **Contractual Risks**: Delays or non-compliance by external vendors or healthcare providers.
- **Technological Risks**: Risks related to system integration, technology stack compatibility, and unforeseen technical challenges.
- **Size and Complexity**: Risks due to the scale of the system and the need to integrate with existing healthcare systems.
- **Personnel Risks**: Challenges in hiring and retaining skilled team members.
- **Customer Acceptance**: Risks in meeting user expectations and gaining acceptance from healthcare staff.

Risk Analysis

- Each identified risk will be analyzed for its potential impact on the project, including likelihood and severity. A risk matrix will categorize risks as **high**, **medium**, or **low**.

Risk Mitigation and Contingency Planning

- **Mitigation Plans**: Develop strategies to reduce the likelihood of risks (e.g., regular training for team members, regular vendor assessments).
- **Contingency Plans**: Define actions to be taken if a risk occurs, such as having backup vendors or additional resources in place.

Tracking Risks

- Risks will be tracked in a **Risk Register**, updated regularly, and reviewed during status meetings to monitor changes and effectiveness of mitigation strategies.

3.4 Monitoring and Controlling Mechanisms

Monitoring and controlling mechanisms ensure the project adheres to the **SPMP** and stays on track in terms of time, budget, and scope. This will include regular reporting, review processes, and quality assurance.

Reporting Mechanisms

- **Status Reports**: Weekly reports will be provided to stakeholders outlining progress, risks, and issues. These reports will include milestones completed, work completed, and upcoming tasks.
- **Risk Reviews**: Monthly review meetings will be held to assess the status of risk factors and update mitigation strategies.
- **Change Control**: A formal change control process will be implemented for managing any scope changes or deviations from the plan.

Audit Mechanisms

- **Internal Audits**: Conduct regular internal audits to ensure project compliance with standards, regulations, and the **SPMP**.
- **Quality Assurance Reviews**: Quality assurance will be integrated into each phase of the project, ensuring that deliverables meet the required standards before proceeding.

Monitoring Tools

- **Project Management Software**: Tools like **Trello**, **JIRA**, or **Microsoft Project** will be used for tracking tasks, milestones, and deadlines.
- **Version Control Systems**: Git or other version control systems will be used for managing code changes and updates.

Frequency	**Report Type**	**Audience**	**Responsible**	**Method**
Weekly	Project Status Report	Project Team, Stakeholders	Project Manager	Email, Slack
Monthly	Risk and Issues Report	Project Team, Stakeholders	Risk Manager	Meeting, Email
As-needed	Change Requests	Project Manager, Stakeholders	Project Manager	Email, Meetings

Table F-4: Communication and Reporting Plan

3.5 Staffing Approach.

The **HMS** project requires skills in **project management**, **software development** (Java, Python), **healthcare compliance** (HIPAA, GDPR), **quality assurance**, and **UI/UX design**. Recruitment will focus on internal resources first, with external hiring or contractors for specialized roles as needed. Team members will be trained on healthcare regulations, relevant software tools, and system integration. **End-user training** will be provided during implementation to ensure smooth adoption. The project will be supported by a **core team** (PM, developers, QA) and **external experts** for healthcare compliance and system integration.

4. Technical Process

This section outlines the methods, tools, techniques, and work products for developing and maintaining the **Healthcare Management System (HMS)**.

4.1 Methods, Tools, and Techniques

- **Development Method**: Agile methodology for iterative development and quick feedback.
- **Programming Languages**: Java (backend), React (frontend), SQL (database management).
- **Tools**:
- **Version Control**: Git (GitHub/GitLab)
- **Project Management**: JIRA, Trello
- **Testing**: Selenium, JUnit
- **Standards**: HIPAA compliance, GDPR for data security, industry-standard coding practices.
- **Team Structure**: Cross-functional teams (developers, QA, UI/UX designers, project managers).

4.2 Software Documentation

- **Software Requirements Specification (SRS)**: Describes functional requirements, performance, and design constraints, with clear methods for verification (e.g., testing, inspection).
- **Software Design Description (SDD)**: Details the system architecture, databases, and interfaces, providing a blueprint for development.

4.2.1 Software Requirements Specification (SRS)

The **SRS** document will clearly define the system’s functional requirements, performance criteria, design constraints, and external interfaces. Each requirement will be measurable and testable, ensuring verification through methods like **inspection**, **analysis**, **demonstration**, or **testing**. For example, the system must support **HIPAA compliance** for data privacy, and each feature will have clear acceptance criteria for validation.

4.2.2 Software Design Description (SDD)

The **SDD** will outline the software's architecture, including the major components, database structures, and internal interfaces. This document will detail how different modules interact and provide the blueprint for development and integration. It will also include descriptions of **data flow**, **component diagrams**, and the **technology stack** used to implement the system.

4.2.3 Software Test Plan

The **Software Test Plan** will define the methods and procedures for testing all aspects of the **HMS**. This includes:

- **Requirement Testing**: Verifying functionality as outlined in the SRS.
- **Design Testing**: Ensuring design components, as described in the SDD, meet the expected architecture and interfaces.
- **Code Testing**: Conducting unit, integration, and system tests on the implemented software. The plan will detail **test cases**, **procedures**, and expected **test results**, and will ensure the software meets quality standards.

4.3 User Documentation

User documentation will include both **online and offline** resources, offering clear guidance for system users. This includes:

- **Online Help**: Contextual help integrated within the application.
- **User Manuals**: Detailed documentation available in PDF format for end-users, including **administrators**, **doctors**, and **nurses**.
- **Support Resources**: A network-accessible knowledge base, FAQs, and troubleshooting guides.

Documentation will be developed in parallel with the system and will be updated throughout the project lifecycle.

4.4 Project Support Functions

Support functions include:

- **Configuration Management**: Ensuring version control and tracking changes in code, documents, and system components.

- **Software Quality Assurance (QA)**: Monitoring the project for adherence to quality standards, testing processes, and ensuring software meets functional and non-functional requirements.
- **Verification and Validation**: Ensuring that all requirements (from the SRS) are met and that the system satisfies customer needs.

Each support function will have specific **responsibilities**, **resource requirements**, **schedules**, and **budgets**. These functions will be critical to ensuring the successful delivery of the **HMS**.

5. Work Packages, Schedule, and Budget

This section outlines the key work packages, their dependencies, required resources, and the project schedule and budget.

5.1 Work Packages

Work packages represent the specific activities and tasks necessary to complete the **HMS** project. Key work packages include:

- **Requirement Gathering**: Define system specifications based on stakeholder needs.
- **System Design**: Develop system architecture, database models, and UI designs.
- **Development**: Implement the system's backend, frontend, and integration components.
- **Testing**: Conduct unit, integration, and system tests to validate the software.
- **Deployment**: Deploy the software and provide post-launch support.

A **Work Breakdown Structure (WBS)** diagram will visually map out these work packages and their hierarchical relationships.

5.2 Dependencies

Dependencies between work packages will be managed using techniques such as:

- **Activity Network**: Visualize tasks and their interdependencies.
- **Critical Path Method**: Identify the longest sequence of dependent tasks to determine project duration. Key dependencies include:
- **Design completion** → Development starts.
- **Development completion** → Testing phase starts.

5.3 Resource Requirements

Resource requirements are estimated based on the project's timeline:

- **Personnel**: Developers (5), QA Engineers (2), UI/UX Designers (2), Project Manager (1)
- **Hardware/Software**: Development servers, testing tools (Selenium, JUnit), databases (SQL Server/MySQL).
- **Facilities**: Office space, remote work setups, meeting rooms.

5.4 Budget and Resource Allocation

The project budget will be allocated across various functions, including:

- **Personnel Costs**: Salaries for development, QA, and management teams.
- **Software Tools**: Licensing for development, testing, and deployment tools.
- **Hardware**: Servers for development and testing environments.
- **Miscellaneous**: Office supplies, travel for team meetings.

A detailed budget breakdown will be updated regularly.

5.5 Schedule

The project schedule will outline key milestones:

- **Phase 1**: Requirements Gathering (Month 1)
- **Phase 2**: System Design (Month 2-3)
- **Phase 3**: Development (Month 4-6)
- **Phase 4**: Testing (Month 7-8)
- **Phase 5**: Deployment and Maintenance (Month 9)

6. Additional Components.

Certain additional components may be necessary to support the project. These sections will be added as needed.

6.1 Index.

Already given in starting of the page .

6.2 Appendices

Appendices may be included, either directly or by reference, to provide supporting details that could detract from the SPMP if included in the body of the SPMP. Suggested appendices include:

A. Current Top 10 Risk Chart :-

This appendix will outline the current top 10 risks identified for the projects ,their likelihood, impact and mitigation strategies

B. Current Project Work Breakdown Structure :-

The Work Breakdown Structure (WBS) will provide a detailed breakdown of project activities, tasks, and their dependencies.

C. Current Detailed Project Schedule :-

This appendix will present a detailed project schedule, including milestones, deadlines, and resource allocation for each work package.

Conclusion:

Through this experiment we understand the how to organize a project and ensure it meets deadlines. We create a project management plan for our mini project – HealthCare Management System.

Post Lab Descriptive Questions

1. State various Scheduling principles and explain them in detail.

Scheduling principles are essential in project management as they help in the effective allocation of resources, time management, and the successful completion of tasks. Here are various scheduling principles and their explanations in detail:

1. **Define Clear Objectives:**

   - **Explanation:** The first step in scheduling is to define clear and specific project objectives. These objectives help in determining the scope of work, expected outcomes, and deliverables. Without well-defined objectives, scheduling becomes challenging as there is no clear direction for the project.

2. **Task Breakdown:**

   - **Explanation:** Break down the project into smaller, manageable tasks or work packages. This division helps in organizing the project and understanding the interdependencies between tasks. A Work Breakdown Structure (WBS) is often used to categorize and structure the tasks.

3. **Sequencing:**

   - **Explanation:** Once tasks are identified, they need to be sequenced in the order they should be executed. Sequencing ensures that tasks are performed in a logical and efficient sequence, taking into account dependencies between them.

4. **Task Duration Estimation:**

   - **Explanation:** Accurately estimate the time required to complete each task. This estimation helps in creating a realistic schedule. Various techniques, such as expert judgment, historical data, and analogous estimation, can be used for task duration estimation.

5. **Resource Allocation:**

   - **Explanation:** Allocate the necessary resources, including personnel, equipment, and materials, to each task. Resource allocation ensures that tasks have the required resources available when needed, preventing bottlenecks and resource constraints.

6. **Critical Path Analysis:**

   - **Explanation:** Identify the critical path, which is the sequence of tasks that determines the project's overall duration. Tasks on the critical path have no slack, meaning any delay in these tasks will delay the project's completion. This analysis is crucial for time-sensitive projects.

7. **Buffer Management:**

   - **Explanation:** Incorporate buffers or contingency time in the schedule to account for uncertainties and unexpected delays. These buffers, often called slack or float, provide flexibility and help mitigate risks that could impact the project timeline.

8. **Task Dependencies:**

   - **Explanation:** Understand and define the dependencies between tasks, which can be categorized as finish-to-start, start-to-start, finish-to-finish, or start-to-finish. Recognizing dependencies is crucial for determining the order in which tasks must be executed.

9. **Resource Leveling:**

- **Explanation:** Ensure that resources are efficiently utilized by resolving resource conflicts and over-allocations. Resource leveling helps in optimizing resource usage and preventing resource burnout.

10. **Monitoring and Control:**

- **Explanation:** Continuously monitor the project's progress against the schedule. Any deviations from the plan should be identified and addressed promptly. Adjustments may need to be made to accommodate changes or unforeseen issues.

11. **Regular Updates:**

- **Explanation:** Schedules should be living documents that are regularly updated to reflect the project's current status. As the project progresses, tasks may be completed earlier or later than planned, and updates ensure that the schedule remains accurate.

12. **Communication and Stakeholder Involvement:**

- **Explanation:** Effective communication with project stakeholders is essential. Keep stakeholders informed about the project's schedule, progress, and any changes. Involving stakeholders in scheduling decisions can lead to better collaboration and buy-in.