

NYCPS TMS: Hyper-Detailed Prescriptive Communications Strategy & Plan

I. Introduction: Principles & Objectives

This document mandates the comprehensive communications strategy for the NYCPS Transportation Management System (TMS) project. Given the project's scale, complexity, geographical distribution of the team, public visibility, and stringent government accountability requirements, this plan emphasizes ****proactive, transparent, data-driven, documented, and audience-tailored communication**** at all levels.

Our approach is integrated with the project's Agile (Scrum) methodology and DevSecOps framework, utilizing GitLab, Jira, and Confluence as central hubs for information, supported by real-time channels like Slack/Teams and formal reporting mechanisms.

Core Communication Principles:

- **Transparency & Openness:** Default to sharing information widely unless restricted by confidentiality (PII, security details). Make project status, risks, and decisions visible.
- **Predictability & Cadence:** Establish a regular rhythm of communication through defined meetings and reports.
- **Audience-Centricity:** Tailor the level of detail, format, and content to the specific needs of each stakeholder group (e.g., executive summaries vs. detailed technical updates).
- **Data-Driven Reporting:** Base status reports and discussions on objective data extracted from project tools (Jira, GitLab CI/CD, CloudWatch) whenever possible. Automate data collection.
- **Proactive Risk & Issue Communication:** Escalate blockers, risks, and requests for help *early* through appropriate channels. No surprises.
- **Single Source of Truth:** Utilize Confluence/SharePoint for official documentation, meeting minutes, decisions, and reports. Use Jira/ADO for tracking work items, risks, and issues.
- **Meticulous Documentation:** Mandate agendas, minutes, and action item tracking (with owners and due dates) for all formal meetings. Archive all reports and official communications.

- **Action-Oriented:** Communications should lead to clear decisions and actionable outcomes.

II. Communication Channels & Tooling Implementation

We will standardize on a core set of tools to facilitate different types of communication.

- **Primary Project Hub (Async, Documentation):**
 - **Confluence / SharePoint:** Official repository for project documentation, meeting agendas/minutes, decision logs, architecture documents, runbooks, KB articles, formal reports.
 - **Implementation:** Set up dedicated project space with clear page hierarchy. Define page templates for meeting minutes, ADRs, reports. Configure permissions based on roles/groups. Ensure regular backups.

**Responsibility: Project Manager, Tech Leads,
All Team Members (Content Creation).**

- **Work & Issue Tracking (Async, Task Status):**
 - **Jira / Azure DevOps (ADO):** Master system for tracking Epics, Features, User Stories, Tasks, Bugs, Risks, Action Items. Source for sprint boards, burndown charts, velocity metrics, defect reports.
 - **Implementation:** Configure project workflows, issue types, custom fields (e.g., for Risk Likelihood/Impact), boards (Scrum boards per team), dashboards (project health, sprint progress), and reporting gadgets. Integrate tightly with GitLab (linking commits/MRs to issues).

**Responsibility: Project Manager, Scrum
Masters, Jira/ADO Administrator.**

- **Code & CI/CD Platform (Async, Technical Status):**
 - **GitLab (Ultimate Recommended):** Source code management, Merge Requests (code reviews), CI/CD pipelines (build/test/scan/deploy status), Container Registry, Package Registry, Security Dashboards, potentially Wiki for dev-focused documentation.
 - **Implementation:** Configure repositories, branch protections, CI/CD pipelines (`.gitlab-

ci.yml`), runners, security scanning features (SAST, DAST, SCA), review/approval workflows as defined in DevSecOps strategy. Utilize GitLab MR comments for code-level discussion.

Responsibility: DevOps Team, Development Teams.

- **Real-Time Team Communication (Sync/Async):**
 - **Slack / Microsoft Teams:** Daily team coordination, quick questions, ad-hoc discussions, automated notifications from CI/CD/monitoring, dedicated channels for teams/features/incidents.
 - **Implementation:** Create channels (e.g., `#tms-announce`, `#tms-general`, `#tms-team-alpha`, `#tms-backend`, `#tms-frontend`, `#tms-ops-alerts`, `#tms-incident-XYZ`). Establish communication etiquette (use threads, minimize noise in main channels). Integrate GitLab, Jira, PagerDuty, CloudWatch alerts via webhooks/bots.

Responsibility: Project Manager, Scrum Masters, All Team Members.

- **Meetings (Synchronous):**
 - **Video Conferencing (Zoom/Teams/Webex):** Primary tool for

scheduled meetings involving globally distributed team members.

- **Implementation:** Ensure all team members have access and necessary hardware (webcam, headset). Record key meetings (e.g., Sprint Reviews, Governance) with consent where appropriate and link recordings in minutes (Confluence). Always use meeting invites with clear agendas.

Responsibility: Meeting Facilitators, Project Management.

- **Formal Reporting & Announcements (Async):**
 - **Email:** Used for formal distribution of major reports (e.g., Weekly/Monthly Status), official project announcements, and communication with external stakeholders without direct tool access.
 - **Confluence/SharePoint:** The canonical location for published reports; email often links here.

Responsibility: Project Manager, Communication Lead (if applicable).

- **Incident Communication (External):**
 - **Status Page (e.g., Statuspage.io, Cachet):** Public-facing page for communicating major

service disruptions or planned maintenance affecting end-users (Parents, Students, School Admins).

- **Implementation:** Configure status page with components representing key services. Define incident templates and communication protocols. Integrate with incident management tools if possible.

Responsibility: SRE/Ops Team, Comms Lead (during incidents).

III. Communication Cadence, Artifacts & Meetings

This section details the specific communication events, reports, and meetings, including their purpose, audience, frequency, content, data sources, automation potential, ownership, and associated templates/checklists.

A. Daily Communications

1. Daily Stand-up Meeting

- **Purpose:** Synchronize team members on progress, plans for the day, and identify immediate blockers. Not a status report meeting for management.
- **Audience:** Development Team QA Engineers Scrum Master
(Optional: PO, DevOps, SRE)
- **Frequency:** Daily
- **Duration:** 15 minutes (strictly timed)
- **Format:** Video Conference (standing encouraged if co-located)
- **Content/Agenda Template:**
 - (Each team member answers) What did I accomplish yesterday?
 - (Each team member answers) What will I work on today?
 - (Each team member answers) What impediments are blocking my progress?
 - Focus on identifying blockers and scheduling necessary follow-up discussions *outside* the stand-up.
- **Data Source(s):** Jira/ADO Board (visual aid), Individual Updates.

- **Automation:** Minimal; potentially bot reminders or integration with Jira board updates.
- **Owner/Facilitator:** Scrum Master (ensures meeting happens and stays focused).
- **Documentation:** Blockers identified are immediately tracked in Jira/ADO or addressed post-meeting. Formal minutes generally not required.

2. Team Chat Channels (e.g., Slack/Teams)

- **Purpose:** Ongoing asynchronous communication, quick questions, sharing links, automated notifications, coordination within/between teams.
- **Audience:** All Project Team Members (within relevant channels)
- **Frequency:** Continuous / As Needed
- **Format:** Chat Platform
- **Content:** Technical questions, coordination requests, CI/CD notifications, monitoring alerts (non-critical), links to documentation/MRs, informal team building.
- **Data Source(s):** User Input, Automated Bots/Webhooks (GitLab, Jira, CloudWatch, PagerDuty).
- **Automation:** Integrate tools for automated notifications.
- **Owner/Facilitator:** All team members (responsible for professional communication), Channel Admins/Leads (channel setup/etiquette).

- **Documentation:** Chat history provides informal log; critical decisions or action items should be formally documented in Jira/Confluence.

B. Sprint-Based Communications (Assuming 2-Week Sprints)

1. Sprint Planning Meeting

- **Purpose:** Define the Sprint Goal and select User Stories from the prioritized Product Backlog for the upcoming sprint. Break stories into tasks.
- **Audience:** Development Team QA Engineers Scrum Master
Product Owner (Mandatory) DevOps/SRE Rep (Optional)
- **Frequency:** Once per Sprint (Start)
- **Duration:** 2-4 hours (for a 2-week sprint)
- **Format:** Video Conference
- **Content/Agenda Template:**
 - Review Product Owner's priorities and proposed Sprint Goal.
 - Review team capacity for the sprint.
 - Discuss and select high-priority User Stories from the backlog that meet Definition of

Ready (DoR).

- Team breaks down selected stories into smaller technical tasks (in Jira/ADO).
- Team estimates tasks (optional, focus on story points).
- Team confirms understanding and commits to the Sprint Backlog and Goal.
- Identify dependencies and potential risks for the sprint.
- **Data Source(s):** Prioritized Product Backlog (Jira/ADO), Team Velocity/Capacity Data, DoR Checklist.
- **Automation:** Jira/ADO board setup facilitates backlog visualization and task creation.
- **Owner/Facilitator:** Scrum Master.
- **Documentation:** Sprint Goal and committed Sprint Backlog documented in Jira/ADO. Meeting decisions/risks captured in brief Confluence notes.

2. Backlog Grooming / Refinement Meeting

- **Purpose:** Prepare User Stories for future sprints. Refine descriptions, clarify acceptance criteria, break down large stories (Epics/Features), estimate effort (Story Points), and ensure stories meet Definition of Ready (DoR).

- **Audience:** Development Team QA Engineers Scrum Master
Product Owner Business Analyst (Tech Leads, Architects invited as needed)
- **Frequency:** 1-2 times per Sprint (Mid-Sprint)
- **Duration:** 1-2 hours
- **Format:** Video Conference
- **Content/Agenda Template:**
 - Review top-priority items in the Product Backlog identified by the Product Owner.
 - Discuss user stories: Clarify scope, dependencies, and acceptance criteria.
 - Split large stories into smaller, manageable ones.
 - Estimate effort using techniques like Planning Poker (Story Points).
 - Identify and address impediments to meeting DoR for upcoming stories.
- **Data Source(s):** Product Backlog (Jira/ADO), Architecture Documents, UX Designs.
- **Automation:** Jira/ADO used for tracking story status and estimates.
- **Owner/Facilitator:** Scrum Master / Product Owner.

- **Documentation:** Updated User Stories (description, acceptance criteria, estimates) in Jira/ADO. Clarifications noted in Confluence or Jira comments.

3. Sprint Review Meeting

- **Purpose:** Demonstrate the potentially shippable product increment completed during the sprint. Gather feedback from stakeholders on the delivered functionality. Celebrate team accomplishments.
- **Audience:** Development Team QA Engineers Scrum Master
Product Owner NYCPS Stakeholders (OPT SMEs, School Reps, DIIT Liaisons, etc.) Project Managers Other Interested Parties
- **Frequency:** Once per Sprint (End)
- **Duration:** 1-2 hours
- **Format:** Video Conference (Live Demo preferred)
- **Content/Agenda Template:**
 - Introduction: Sprint Goal recap, attendees.
 - Demo: Team demonstrates *completed* user stories (meeting DoD) using the deployed software (typically in QA or Staging environment). Avoid PowerPoint slides for demos.
 - Q&A / Feedback Session: Stakeholders provide feedback and ask questions.

- Discussion: Product Owner discusses progress towards release goals, backlog updates, potential next steps.
- (Optional) Review key metrics (Velocity, Burndown).
- **Data Source(s):** Completed Sprint Backlog (Jira/ADO), Deployed software increment.
- **Automation:** Demo environment readiness automated via CD pipeline.
- **Owner/Facilitator:** Scrum Master (facilitates), Development Team (conducts demo), Product Owner (leads discussion).
- **Documentation:** Meeting minutes captured in Confluence (attendees, feedback summary, decisions). Product Owner formally accepts/rejects stories in Jira/ADO. New backlog items created based on feedback.

4. Sprint Retrospective Meeting

- **Purpose:** Inspect the team's process during the last sprint and identify actionable improvements for the next sprint. Focus on collaboration, tools, processes, and team dynamics. Blameless environment.
- **Audience:** Development Team QA Engineers Scrum Master (Product Owner optional but encouraged)
- **Frequency:** Once per Sprint (End, after Sprint Review)
- **Duration:** 1 - 1.5 hours

- **Format:** Video Conference (Utilizing collaborative tools like Miro or digital retro boards)
- **Content/Agenda Template (Example - Start/Stop/Continue):**
 - Set the Stage: Review sprint goal, check-in.
 - Gather Data: Brainstorm items for "What should we START doing?", "What should we STOP doing?", "What should we CONTINUE doing?".
 - Generate Insights: Group related items, discuss patterns and root causes.
 - Decide What To Do: Identify 1-3 concrete, actionable improvement items for the team to focus on in the next sprint. Assign owners if necessary.
 - Close: Summarize actions, appreciate contributions.
- **Data Source(s):** Team member input, Sprint metrics (Burndown, Velocity, CI/CD failures).
- **Automation:** Digital retrospective tools can facilitate brainstorming and grouping.
- **Owner/Facilitator:** Scrum Master.
- **Documentation:** Summary of key discussion points and actionable improvement items documented in Confluence and

potentially added as tasks in Jira/ADO for the next sprint.

C. Weekly Communications

1. Weekly Project Status Meeting

- **Purpose:** Provide a regular sync point for core project leadership (Vendor & NYCPS) to review overall progress, address cross-team dependencies, escalate risks/issues, and make tactical decisions.
- **Audience:** Project Managers (Vendor & NYCPS)
Technical Leads (Vendor & NYCPS) Product Owner(s)
Scrum Masters (Key SMEs/Architects as needed)
- **Frequency:** Weekly
- **Duration:** 60 minutes
- **Format:** Video Conference
- **Content/Agenda Template:**
 - Review Action Items from previous meeting.
 - Overall Project Status Summary (RAG - Red/Amber/Green).
 - Key Accomplishments This Week (Highlights per team/phase).

- Priorities for Next Week.
- Review Key Metrics (Sprint Velocity trends, CI/CD health, High-priority Defects).
- Risk & Issue Review (New risks/issues, updates on mitigation/resolution plans for existing high-priority items).
- Blocker Discussion & Escalation.
- Upcoming Milestones/Dependencies.
- Decisions Needed.
- Action Item Recap.
- **Data Source(s):** Jira/ADO (Sprint status, Risks, Issues), GitLab CI/CD (Pipeline status), Confluence (Previous Minutes).
- **Automation:** Jira/Confluence macros can pull in issue lists/statuses. Dashboards can provide metric snapshots.
- **Owner/Facilitator:** Project Manager (Vendor or NYCPS, alternating or shared).
- **Documentation:** Formal meeting minutes published on Confluence within 24 hours, including attendees, key decisions, discussion points, and action items tracked in Jira/ADO with owners/due dates.

2. Weekly Status Report (Written)

- **Purpose:** Provide a concise, written summary of project status, progress, risks, and issues to a broader group of stakeholders, including leadership who may not attend the weekly meeting.
- **Audience:** Project Team Project Leadership
Steering Committee Key NYCPS Stakeholders
- **Frequency:** Weekly (Published end of week, covering that week)
- **Format:** Confluence Page / Email Digest (linking to Confluence)
- **Content Template:**
 - **Overall Status:** RAG Status
(Red/Amber/Green) with brief justification.
 - **Period Covered:** [Week Start Date] -
[Week End Date]
 - **Key Accomplishments This Week:** Bullet points highlighting significant progress (features completed, milestones met, major risks mitigated).
 - **Key Activities Planned Next Week:**
Bullet points outlining upcoming priorities and focus areas.
 - **Key Metrics Snapshot:**
 - Sprint Velocity
(Current/Average).

- CI/CD Pipeline Success Rate (%).
 - Open Critical/High Defects Count (Trend).
 - Deployment Frequency (if tracking).
- **Risks & Issues Update:** Summary of Top 3-5 critical risks/issues, their status, mitigation/resolution plan, and owner. Link to full Risk/Issue Register in Jira/Confluence.
- **Help Needed / Escalations:** Clear articulation of any blockers requiring intervention from leadership or other teams.
- **Data Source(s):** Jira/ADO (Sprint data, Risks, Issues), GitLab CI/CD (Metrics), Weekly Status Meeting Minutes, Input from Leads.
- **Automation Strategy:**
 - Utilize Jira/ADO REST APIs and GitLab APIs to automatically pull metrics (Velocity, Defect Counts, Pipeline Success Rates).
 - Develop script (e.g., Python) or use Confluence macros to query APIs and populate sections of the report template on Confluence.

- Project Manager finalizes qualitative sections (Accomplishments, Plans, Risk Summaries) based on meeting inputs and tool data.
- **Owner/Facilitator:** Project Manager (Vendor or NYCPS).
- **Documentation:** Published report archived in Confluence project space. Email sent with summary and link.

D. Monthly / Bi-Monthly Communications

1. Monthly Business Review (MBR) / Steering Committee Meeting

- **Purpose:** Provide executive leadership and key business stakeholders with a strategic overview of project progress, alignment with business goals, budget status, major risks, and key decisions needed. Forum for high-level governance and direction setting.
- **Audience:** Steering Committee Members (NYCPS & Vendor Execs)
 Project Sponsors Product Owner(s) Project Managers
 Key Technical/Business Leads (Presenting specific topics)
- **Frequency:** Monthly (or Bi-Monthly, depending on Sponsor preference)
- **Duration:** 60-90 minutes

- **Format:** Video Conference / In-Person (Presentation + Discussion)
- **Content/Agenda Template:**
 - Project Vision & Goals Recap (Briefly)
 - Executive Summary (Key highlights, overall status RAG).
 - Progress Against Roadmap & Key Milestones (Visual timeline showing completed, in-progress, upcoming).
 - Financial Summary (Budget vs. Actual spend, forecast).
 - Key Metrics Review (Focus on trends: DORA metrics, SLO adherence, Defect trends, User Adoption if applicable).
 - Major Risks & Issues (Strategic risks needing executive attention/mitigation support).
 - Upcoming Phase/Release Overview.
 - Key Decisions Required from Steering Committee.
 - Open Discussion / Q&A.
 - Action Item Review.

- **Data Source(s):** Aggregated Weekly Status Reports, Jira/ADO Dashboards, Financial Tracking System, GitLab Insights (DORA metrics), Monitoring Tools (SLO data).
- **Automation Strategy:** Leverage automated data collection for metrics. Dashboards provide live views. PM synthesizes data into executive-level presentation format.
- **Owner/Facilitator:** Project Sponsor / Project Manager.
- **Documentation:** Formal presentation deck and detailed meeting minutes (decisions, action items with owners/dates) published on Confluence and distributed via email.

Governance Gate: Steering Committee provides strategic direction, approves major scope/budget changes, and confirms continued alignment with business objectives.

2. Technical Review Board (TRB) Meeting

- **Purpose:** Review and approve significant architectural decisions, technology choices, security designs, and integration strategies. Ensure technical soundness, scalability, maintainability, and alignment with enterprise standards.
- **Audience:** Cloud Architects Security Architects
Senior Technical Leads (Relevant Domains) DIIT Representatives
SRE/Ops Lead
- **Frequency:** As Needed (triggered by major design proposals), potentially regular cadence (e.g., Bi-Monthly) for proactive review.
- **Duration:** 60-90 minutes per topic

- **Format:** Video Conference
 - **Content/Agenda Template:**
 - Presentation of proposed design/architecture change (Problem statement, proposed solution, alternatives considered, rationale, security implications, operational impact).
 - Supporting documentation (Diagrams, ADRs).
 - Q&A and Technical Discussion.
 - Decision (Approve, Approve with Conditions, Reject, Request More Info).
 - **Data Source(s):** Architecture Documents, Design Documents, ADRs, Threat Models (All stored in Confluence/Git).
 - **Automation:** N/A (Process-driven).
 - **Owner/Facilitator:** Lead Architect / Proposer of the change.
 - **Documentation:** Meeting minutes and formal decision record (including rationale and conditions) documented in Confluence and linked to relevant design documents/ADRs.
- Governance Gate:** Formal sign-off from TRB required before implementing major architectural changes or adopting new core technologies.

3. Security Review Board Meeting

- **Purpose:** Review and approve security architecture, threat models, security control implementations, penetration test results, and risk mitigation plans. Ensure compliance with NYCPS/NYC3 policies and regulatory requirements.
- **Audience:** Security Architects (Vendor & NYCPS/NYC3)
Compliance Officer Technical Leads Project Manager
- **Frequency:** As Needed (triggered by security design completion, test results), potentially regular cadence.
- **Duration:** 60-90 minutes per topic
- **Format:** Video Conference
- **Content/Agenda Template:** Similar to TRB, focused on security aspects (Threat Model review, Control Design validation, Pen Test Findings review, Risk Acceptance requests).
- **Data Source(s):** Security Design Docs, Threat Models, SAST/DAST/SCA/Pen Test Reports, Compliance Checklists.
- **Automation:** Security scanning tools provide input data.
- **Owner/Facilitator:** Lead Security Architect / Project Manager.
- **Documentation:** Meeting minutes and formal security approvals/risk acceptance documented in Confluence. Findings tracked in Jira/ADO.

Governance Gate: Formal sign-off from Security Review Board required for security architecture, control implementation, and before releasing with known vulnerabilities (risk acceptance).

E. Phase-Based Communications

1. Phase Gate Reviews

- **Purpose:** Formal review at the end of major project phases (as defined in Project Plan, e.g., Foundation Complete, Core Modules Complete, UAT Complete) to assess if phase exit criteria have been met and approve progression to the next phase.
- **Audience:** Steering Committee Project Managers
Product Owner Key Technical/QA/Security Leads
- **Frequency:** End of Each Major Project Phase
- **Duration:** 90-120 minutes
- **Format:** Video Conference / In-Person (Formal Presentation)
- **Content/Agenda Template:**
 - Phase Goals & Objectives Recap.
 - Summary of Deliverables Completed during the phase.

- Demonstration of key capabilities achieved (if applicable).
 - Review of Phase Exit Criteria Checklist (Verification).
 - Summary of Testing Results (QA, UAT, Perf, Sec).
 - Review of outstanding risks/issues relevant to proceeding.
 - Plan & Readiness for Next Phase.
 - Go/No-Go Decision for proceeding to next phase.
-
- **Data Source(s):** Project Plan, Phase Exit Criteria Checklist, Test Summary Reports, Jira/ADO (Deliverables status, Risks/Issues).
 - **Automation:** Dashboards provide status, but review requires synthesis and presentation.
 - **Owner/Facilitator:** Project Manager.
 - **Documentation:** Formal Phase Gate Review presentation deck and meeting minutes with Go/No-Go decision recorded in Confluence.
-
- Governance Gate:** Formal Go/No-Go decision required from Steering Committee/Sponsors to proceed to the next project phase.

F. Ad-hoc & Incident Communications

1. Blocker/Impediment Resolution

- **Purpose:** Quickly address issues blocking team progress identified during Daily Stand-ups or other channels.
- **Audience:** Relevant Team Members Scrum Master (Potentially Leads/PMs if escalation needed)
- **Frequency:** As Needed
- **Format:** Short Video Call / Targeted Chat / In-Person Huddle
- **Content:** Focused discussion on the specific blocker and agreement on next steps/ownership for resolution.
- **Documentation:** Blocker status updated in Jira/ADO. Resolution path briefly noted if significant.

2. Incident Communication (Production Issues)

- **Purpose:** Manage communication effectively during production incidents to facilitate rapid resolution and keep stakeholders informed.
- **Audience (Internal):**
 - Incident Response Team (On-call SRE/Ops, SMEs)
 - Incident Commander Comms Lead Project Leadership
- **Audience (External/Stakeholder):**
 - NYCPS Leadership/Stakeholders Support Teams (L1/L2)
 - End Users (via Status Page for major incidents)

- **Frequency:** Continuous during Incident (Internal),
Regular Updates (External - e.g., every 15-60 mins depending on severity/impact)
- **Format:** Dedicated Slack/Teams Channel (Internal War Room)
Video Bridge (Internal) Status Page (External)
Email Summaries (Stakeholders)
- **Content:**
 - **Internal:** Real-time updates on investigation, hypotheses, actions taken, impact assessment.
 - **External (Status Page):** Acknowledge issue, describe user impact (briefly, non-technically), provide ETA if known (use caution), post updates on investigation/resolution progress, confirm resolution.
 - **Stakeholder Updates:** Summaries of impact, status, ETA, and business implications.
- **Implementation:** Follow predefined Incident Management Process (see DevSecOps Strategy). Use PagerDuty/Opsgenie for initial alerting/engagement. Utilize Statuspage.io (or similar) for external updates. Define communication templates for different incident stages/severities.

Responsibility: Incident Commander (overall coordination), Comms Lead (stakeholder/external

comms), On-Call SRE/Ops, SMEs.

- **Documentation:** Incident timeline, decisions, actions meticulously documented in Incident Ticket (Jira/ServiceNow) and summarized in Post-Mortem report (Confluence).

3. Ad-hoc Technical Discussions / Design Sessions

- **Purpose:** Collaborative problem-solving, design brainstorming, knowledge sharing.
- **Audience:** Relevant Developers, Leads, Architects, QA, Ops
- **Frequency:** As Needed
- **Format:** Video Conference / Whiteboarding Session (Virtual/Physical)
- **Documentation:** Key decisions or designs resulting from ad-hoc discussions should be captured briefly in Confluence, ADRs, or Jira tickets to ensure persistence.

G. Documentation & Archival Strategy

1. Central Repository

- **Action:** Establish Confluence (or SharePoint) as the single source of truth for all project documentation (non-code artifacts). Create a well-defined space/site structure (e.g., by Phase, by Topic Area - Requirements, Design, Testing, Operations, Meetings).

- **Action:** Use consistent naming conventions and labeling/tagging for easy searching.
- **Action:** Configure appropriate permissions for read/write access based on roles.

Responsibility: Project Manager, Tech Leads, Confluence Admin.

2. Meeting Documentation

- **Action:** Mandate use of standard Agenda and Minutes templates for all formal meetings (Stand-ups excluded from formal minutes). Templates stored in Confluence.
- **Action:** Minutes must be published to Confluence within 24-48 hours of the meeting conclusion.
- **Action:** Action items identified in meetings must be captured in the minutes AND created/linked as trackable tasks in Jira/ADO, assigned to an owner with a due date.

Responsibility: Meeting Facilitator/Scrum Master/Project Coordinator.

3. Reporting Archival

- **Action:** All formal reports (Weekly Status, Monthly Business Review decks, Test Summary Reports, Security Reports, Phase Gate Reviews) must be stored/published on the designated Confluence space.
- **Action:** Maintain version history for key documents.

Responsibility: Report Owners (PMs, QA Lead, Sec Lead).

4. Decision Log

- **Action:** Maintain a central Decision Log in Confluence capturing key technical, architectural, and project decisions, date made, rationale, stakeholders involved, and links to supporting documentation/minutes.

Responsibility: Project Manager / Tech Leads.

5. Audit Trail

- **Action:** Leverage built-in audit trails within GitLab (commits, MRs, pipeline history), Jira (issue history), and Confluence (page history) to track changes.
- **Action:** Ensure AWS CloudTrail is configured correctly for infrastructure change auditing.

Responsibility: DevOps Team, Tool Administrators.

X. Conclusion

This comprehensive Communications Plan provides the framework for ensuring effective, transparent, and documented communication throughout the NYCPS TMS project lifecycle. Adherence to the defined cadences, templates, tools, and responsibilities is crucial for managing stakeholder expectations, mitigating risks, facilitating collaboration within a large distributed team, and meeting the

stringent accountability requirements inherent in this critical public sector project. Continuous refinement of this plan based on project experience and feedback will be essential for ongoing success.