NYCPS TMS: Formal Change & Risk Management Plan

I. Introduction & Integration with Agile/DevOps

This document defines the formal Change Management and Risk Management processes supplementing the core Agile (Scrum) and DevSecOps practices for the NYCPS Transportation Management System (TMS) project. While Agile methodologies embrace iterative refinement within sprints based on feedback and backlog grooming, the scale, complexity, contractual nature, and public sector context of this project necessitate a formal structure for managing significant changes to baselined scope, schedule, or budget, and for proactively managing project risks.

Integration Principle: These formal processes are designed to work *with*, not against, the Agile workflow. They provide a framework for control and

accountability for substantial deviations or potential threats, while day-to-day development and refinement continue within the sprint cadence.

- Approved Change Requests (CRs) translate into prioritized Epics/Features/User Stories within the Product Backlog.
- Risk Mitigation actions become tasks within the Sprint Backlog.
- Regular Agile ceremonies (Reviews, Retrospectives, Grooming) serve as inputs for identifying potential changes and risks.
- Automation via Jira/ADO and Confluence streamlines tracking and reporting for both Agile execution and formal PM processes.

Adherence to these Change and Risk Management procedures is mandatory for all project team members and stakeholders to ensure controlled execution, transparency, and successful delivery against commitments.

II. Governance Context for Change & Risk

The effectiveness of these plans relies on the established project governance structure. Key bodies involved include:

- **Project Managers (Vendor & NYCPS):** Co-manage day-to-day execution, facilitate processes, perform initial CR/Risk assessment, prepare reports.

- **Product Owner(s) (NYCPS):** Owns the Product Backlog, key decision-maker for functional scope changes within Agile framework, provides input on CR impact/value.
- **Change Control Board (CCB):** Formal body (membership defined: e.g., PMs, PO, Key Tech Leads, Sponsor Rep) responsible for reviewing and approving/rejecting significant Change Requests impacting baseline scope, schedule, or cost. Meets regularly (e.g., bi-weekly) or ad-hoc as needed.
- **Technical Review Board (TRB):** Reviews technical impact of changes, assesses technical risks.
- **Security Review Board (SRB):** Reviews security implications of changes, assesses security risks.
- **Steering Committee:** Provides highest-level oversight, resolves escalated risks/issues, approves major strategic changes or budget adjustments recommended by the CCB.

III. Detailed Change Management Plan

A. Purpose & Scope

Purpose: To provide a structured, documented, and controlled process for managing requests to change the agreed-upon project scope, schedule, cost, or technical approach *after* the initial requirements baseline for a major phase or release has been formally approved (typically post Phase 1 requirements sign-off).

Scope - What Requires a Formal Change Request (CR):

- Requests for significant new features or functionality not included in the baselined requirements for the current or upcoming planned release/phase.
- Changes that materially impact the project schedule (e.g., delay a major milestone by more than one sprint) or budget (e.g., require additional resources or licenses).
- Changes impacting contractual obligations or deliverables.
- Significant alterations to the approved technical architecture or core security controls.
- Changes requested by NYCPS stakeholders *after* UAT sign-off for a specific release increment.

Out of Scope (Handled via Agile Backlog Management):

- Clarifications or refinements of existing user stories during backlog grooming or sprint execution.
- Bug fixes for implemented functionality (managed via defect tracking).
- Minor technical implementation details decided by the development team that do not impact external interfaces,

NFRs, or overall architecture.

- Reprioritization of *existing* backlog items by the Product Owner (unless it significantly impacts major milestones requiring CCB notification).

B. Change Request (CR) Workflow & Process Steps

1. **Step 1: CR Initiation & Submission**

   Description:

   Any project stakeholder (Vendor team member, NYCPS stakeholder) identifies a potential need for change that falls within the scope defined above.

   Implementation How-To:

   a. Requestor creates a new issue of type "Change Request" in the project's Jira/ADO board.

   b. Requestor completes the mandatory fields defined in the **Change Request Form Template** (see below). This

includes a clear description, strong justification/business value, requestor details, date, and initial thoughts on impact/priority.

c. Attach any supporting documentation (e.g., mockups, requirement documents).

Responsibility: Any Stakeholder (Initiation), Requestor (Documentation).

2. **Step 2: CR Logging & Initial Assessment**

Description:

The Project Manager (or designated Change Manager) reviews the submitted CR for completeness and logs it formally.

Implementation How-To:

a. Project Manager reviews the submitted Jira ticket for clarity and required information within 1-2 business days.

b. If incomplete, PM requests clarification from the Requestor via Jira comments.

c. If complete, PM assigns a unique CR ID (can be the Jira key), updates the status

to "Under Assessment", and adds it to the central Change Register (maintained as a Jira Filter/Dashboard and potentially summarized on a Confluence page).

d. PM performs a quick initial assessment: Does this clearly require CCB review, or could it potentially be addressed as a backlog refinement/bug? Assigns initial priority estimate.

Responsibility: Project Manager.

3. **Step 3: Detailed Impact Assessment**

Description:

Relevant Subject Matter Experts (SMEs) conduct a thorough analysis of the proposed change's impact across multiple dimensions.

Implementation How-To:

a. PM assigns assessment tasks to relevant SMEs via linked Jira tasks or comments (e.g., Tech Lead for technical feasibility/effort, QA Lead for testing impact, Security Architect for security implications, PO for value/priority reassessment, PM for schedule/cost).

b. SMEs perform analysis within an agreed timeframe (e.g., 3-5 business days).

c. SMEs document their findings using the **Impact Assessment Template** (see below), typically on a linked Confluence page or within dedicated Jira custom fields. Analysis must cover:

    - **Scope:** How does this change functionality? What other areas/stories are affected?
    - **Schedule:** Estimated effort (e.g., Story Points, Person-Days), impact on current sprint, impact on major milestones/release dates.
    - **Cost:** Additional resource needs, software/licensing costs, AWS GovCloud cost implications.
    - **Technical:** Feasibility, architectural

impact, complexity, impact on performance/scalability/reliability.

- **Testing:** Additional testing required (automated/manual), impact on existing test suites.
- **Security:** Potential new vulnerabilities, impact on existing controls, compliance implications.
- **Risk:** New risks introduced, impact on existing risks.
- **Dependencies:** Impact on or from other teams/systems/features.

Responsibility: Assigned SMEs (Tech Lead, QA Lead, Sec Lead, PO, PM, Arch, etc.).

4. **Step 4: PM Review & Recommendation**

Description:

The Project Manager consolidates the impact assessment and formulates a recommendation for the CCB.

Implementation How-To:

a. PM reviews the completed impact assessments for thoroughness and consistency.

b. PM synthesizes the findings into a summary.

c. PM develops a formal recommendation: Approve, Reject, Defer, or Request More Information, including rationale based on the impact analysis vs. the original justification/value.

d. PM updates the CR status in Jira/ADO to "Ready for CCB Review" and adds it to the agenda for the next CCB meeting.

Responsibility: Project Manager.

5. **Step 5: Change Control Board (CCB) Review & Decision**

Description:

The formally constituted CCB reviews the CR, impact assessment, and recommendation to make a final, documented decision.

Implementation How-To:

a. CCB meeting held at regular cadence (e.g., bi-weekly) or convened ad-hoc for urgent requests. Agenda circulated beforehand.

b. PM or Requestor presents the CR, justification, and summary of impact assessment.

c. SMEs present details of impact assessment as needed.

d. CCB discusses benefits, costs, risks, and alignment with project goals/constraints.

e. CCB makes a formal decision:

- **Approved:** Change proceeds to implementation.
- **Approved with Conditions:** Change proceeds but specific conditions must be met.

- **Rejected:** Change will not be implemented. Rationale documented.
- **Deferred:** Decision postponed pending more information or strategic timing. CR placed on hold.
- **More Information Required:** CR returned to Impact Assessment stage with specific questions.

f. Decision and rationale are formally documented in CCB meeting minutes (Confluence) and the CR status/resolution updated in Jira/ADO.

Responsibility: CCB Chairperson, CCB Members, PM (Presentation/Documentation).

Governance Gate: Formal CCB approval required for any significant changes impacting baselined scope, schedule, or budget.

6. **Step 6: Implementation & Communication**

Description:

If approved, the change is integrated into the project plan and backlog; the decision is communicated.

Implementation How-To:

a. PM updates project schedule, budget forecast, and potentially risk register based on the approved change.

b. Product Owner works with the BA/Team to translate the approved CR into well-defined Epics/Features/User Stories in the Product Backlog.

c. Prioritize the new backlog items according to standard backlog refinement process.

d. PM communicates the CCB decision and its implications (updated schedule, added features) to the wider project team and relevant stakeholders via meeting minutes summary and status reports.

e. Update the Change Register with the final decision and link to implementation backlog items.

Responsibility: Project Manager, Product Owner, Business Analyst, Scrum Master.

C. Change Management Tools & Artifacts

- **Change Request Tracker:** Jira/ADO Project configured with:
    - Custom Issue Type: "Change Request".
    - Custom Workflow: Submitted -> Under Assessment -> Ready for CCB -> CCB Review -> Approved / Rejected / Deferred -> Implemented / Closed.
    - Custom Fields: CR ID, Requestor, Date Submitted, Justification, Business Value, Priority (Initial/Final), Impact Summaries (Scope, Schedule, Cost, Tech, Sec, Risk), CCB Decision, Decision Date, Link to Implementation Tickets, Link to Confluence Impact Doc.
- **Change Register:** A saved Filter/Dashboard in Jira/ADO showing all "Change Request" issues and their current status. Potentially summarized on a Confluence page.
- **Confluence Space:** Dedicated area for:

- Change Management Plan (this document).
- CCB Charter & Membership.
- CCB Meeting Minutes Archive.
- Detailed Impact Assessment Documents (using template).

- **Change Request Form Template (Conceptual - implemented as Jira fields):**

  CR Template Fields:

  - CR ID: (Auto-generated or Jira Key)
  - Title: (Concise summary)
  - Requestor:
  - Date Submitted:
  - Change Description: (Detailed 'What')
  - Justification / Business Value: (Detailed 'Why', benefits, alignment with goals)
  - Proposed Solution (if known):
  - Urgency/Requested Implementation Date:
  - Initial Impact Assessment (Requestor's view – High/Med/Low on Scope/Sched/Cost):

- Affected Components/Modules:
- Attachments:

- **Impact Assessment Template (Conceptual - implemented on Confluence page linked to Jira CR):**

  Impact Assessment Template Sections:

  - CR ID & Title:
  - Assessment Date:
  - Assessor(s) / Role(s):
  - Scope Impact Analysis: (Detailed changes to functionality, interfaces, data)
  - Schedule Impact Analysis: (Estimated effort – Story Points/Days, impact on sprint/milestones)
  - Cost Impact Analysis: (Resource needs, licensing, infra costs)
  - Technical Impact Analysis: (Feasibility, complexity, architectural changes, performance/scalability effects)
  - Testing Impact Analysis: (New test cases needed, automation effort, regression scope)
  - Security Impact Analysis: (New threats/vulnerabilities, control

```
    effectiveness impact, compliance
    check)
  - Risk Impact Analysis: (New risks
    created, existing risks impacted)
  - Dependency Impact Analysis:
    (Impact on/from other
    teams/systems)
  - Alternatives Considered
    (Optional):
  - Overall Assessment Summary &
    Recommendation:
```

D. Roles & Responsibilities Summary (Change Management)

- **Requestor:** Initiates CR, provides justification.
- **Project Manager:** Logs CR, facilitates impact assessment, prepares recommendation, manages Change Register, communicates decisions, updates plans.
- **SMEs (Tech, QA, Sec, PO, etc.):** Perform detailed impact analysis in their area of expertise.
- **Change Control Board (CCB):** Reviews significant CRs, makes Approve/Reject/Defer decisions based on overall

project goals and constraints.

- **Product Owner:** Assesses business value/priority, incorporates approved changes into backlog.
- **Development Team:** Implements approved changes via backlog items.

IV. Detailed Risk Management Plan

A. Purpose & Scope

Purpose: To systematically identify, analyze, plan responses for, monitor, and control potential events or conditions (risks) that could negatively impact the successful achievement of the NYCPS TMS project objectives (scope, schedule, budget, quality, security, stakeholder satisfaction).

Scope: This plan covers all phases of the project lifecycle, from initiation through operations. Risks can be internal (e.g., technical challenges, resource constraints, process failures) or external (e.g., changing regulations, vendor dependencies, infrastructure outages).

B. Risk Management Methodology & Process

1. **Step 1: Risk Identification (Continuous)**

Description:

Proactively and continuously identify potential risks throughout the project.

Implementation How-To:

a. **Initial Risk Workshop:** Conduct a dedicated workshop early in the project (Phase 1) with key stakeholders (PMs, Leads, PO, OPT/DIIT SMEs, Security) using brainstorming techniques.

b. **Ongoing Identification Techniques:**

- **Checklist Analysis:** Use checklists based on common project risks, AWS GovCloud considerations, complex system integrations, and past NYCPS project lessons learned.

- **Assumption Analysis:** Review key project assumptions and identify risks if assumptions prove false.
- **SWOT Analysis:** Periodically analyze Strengths, Weaknesses, Opportunities, Threats related to the project.
- **Agile Ceremonies:** Encourage risk identification during Daily Stand-ups (impediments), Sprint Reviews (feedback indicating unmet needs), Sprint Retrospectives (process issues), and Backlog Grooming (unclear requirements).
- **Expert Interviews:** Discuss potential risks

with technical architects, security specialists, and domain experts.

- **Document Review:** Analyze RFP, contracts, designs, plans for potential risks.

c. **Logging Risks:** Any identified potential risk is logged immediately as a "Risk" issue type in Jira/ADO, including a clear description, potential impact, and identifier.

Responsibility: All Team Members & Stakeholders (Identification), PM/Scrum Master (Facilitation/Logging).

2. Step 2: Qualitative Risk Analysis

Description:

Assess the likelihood (Probability) and potential impact of identified risks to prioritize them.

Implementation How-To:

a. For each logged risk in Jira/ADO, the PM and relevant SMEs assess:

- **Probability (P): Estimate the likelihood of the risk occurring using a defined scale (e.g., 1-Very Low, 2-Low, 3-Medium, 4-High, 5-Very High). Define criteria for each level (e.g., High = >75% chance).**
- **Impact (I): Estimate the potential negative effect on project objectives (schedule, cost, scope, quality, security) if the risk occurs, using a defined scale (e.g., 1-Very Low, 2-Low, 3-Medium, 4-High, 5-Very High). Define criteria (e.g., High Impact = >1 month schedule delay OR**

>10% budget increase OR critical functionality failure OR significant security breach).

b. Calculate **Risk Score/Level: Use a Probability/Impact (PxI) Matrix (see template below) to determine the overall risk level (e.g., Critical, High, Medium, Low) based on P and I ratings. Store P, I, and Score/Level in Jira custom fields.**

c. Prioritize risks based on their Score/Level for response planning.

Responsibility: Project Manager, Relevant SMEs (Tech Lead, QA Lead, Sec Lead, PO).

Probability/Impact Matrix Example:

Probability ↓ / Impact →	1 (VL)	2 (L)	3 (M)	4 (H)	5 (VH)
5 (VH)	M	H	H	C	C
4 (H)	L	M	H	H	C

Probability ↓ / Impact →	1 (VL)	2 (L)	3 (M)	4 (H)	5 (VH)
3 (M)	L	L	M	H	H
2 (L)	L	L	L	M	M
1 (VL)	L	L	L	L	M

(L=Low, M=Medium, H=High, C=Critical - Define specific score ranges for each level)

3. Step 3: Quantitative Risk Analysis (Optional)

Description:

For select Critical or High-impact risks where more data is available, perform numerical analysis of the potential effect.

Implementation How-To:

a. Only performed if deemed necessary by PM/Steering Committee for specific high-stakes risks (e.g., major schedule delays with financial penalties, significant budget overruns).

b. Techniques might include:

- **Expected Monetary Value (EMV): EMV = Probability (%) * Cost Impact ($). Used to compare cost-effectiveness of response strategies.**
- **Monte Carlo Simulation: Use software to model uncertainty in task durations or costs based on probability distributions to forecast overall project schedule/cost ranges.**

Responsibility: Project Manager, Specialized Analyst (if needed).

4. Step 4: Risk Response Planning

Description:

Develop strategies and specific actions to address identified risks, prioritizing based on risk

score.

Implementation How-To:

a. For each Critical and High risk (and Medium risks as appropriate), determine the optimal response strategy:

- ****Avoid:** Change the project plan to eliminate the risk entirely (e.g., remove a high-risk feature, change a technical approach).**
- ****Mitigate:** Take actions to reduce the Probability or Impact (e.g., add extra testing for complex modules, conduct security training, build redundancy, prototype high-risk tech early).**

- **Transfer:** Shift risk to a third party (rare in this context, potentially specific insurance or contractual clauses with hardware suppliers).
- **Accept:** Consciously decide to take no action, typically for Low risks or where response cost outweighs potential impact. **Contingency Planning** is often required for accepted Medium/High risks: Define trigger conditions and specific actions to take *if* the risk occurs.

b. **Document the chosen strategy and detailed response plan (specific actions, steps) within the Jira/ADO Risk ticket or linked Confluence page.**

c. **Assign a specific **Risk Owner** responsible for overseeing the implementation of the response plan.**

d. **If mitigation/avoidance actions require development effort, create corresponding User Stories/Tasks in the Product Backlog and prioritize appropriately.**

Responsibility: Risk Owner, Project Manager, Relevant SMEs, Product Owner (for backlog integration).

5. **Step 5: Risk Monitoring & Control (Continuous)**

Description:

Track identified risks, monitor trigger conditions, implement response plans, identify new risks, and communicate status.

Implementation How-To:

a. **Risk Register Review:** Review the Risk Register (Jira/ADO query/dashboard) in:
    - **Weekly Project Status Meetings:** Discuss status updates for top High/Critical risks, review progress on response actions, identify any newly emerged risks.
    - **MBR/Steering Committee Meetings:** Present summary of key strategic risks, required decisions, or escalations.
    - **Sprint Retrospectives:** Use as a forum to identify new process-related or team-related risks.

b. **Risk Owner Responsibility:** Risk Owners actively monitor their assigned risks and trigger conditions, implement planned responses, and report status changes to the PM.

c. **Re-assessment:** Periodically re-assess probability and impact of existing risks as the project progresses and more information becomes available. Update Risk Register accordingly.

d. **New Risk Identification:** Continuously log new risks as they are identified through any channel.

e. **Risk Closure:** Formally close risks in the register when they are no longer relevant, have occurred (becoming issues), or mitigation is fully implemented and verified.

Responsibility: Project Manager (overall process), Risk Owners (specific risks), All Team Members (ongoing identification).

C. Risk Management Tools & Artifacts

- **Risk Management Plan:** This document (stored in Confluence). Defines methodology, roles, scales, matrix, reporting.
- **Risk Register:** The central artifact. Implemented as:
  - **Preferred:** Jira/ADO Project configured with:
    - **Custom Issue Type: "Risk".**
    - **Custom Workflow: Open -> Analyzing -> Responding (Mitigate/Avoid/Transfer) / Monitoring (Accept) -> Closed / Occurred.**
    - **Custom Fields: Risk ID, Description, Category, Date Identified, Identified By, Probability (Dropdown/Number), Impact**

(Dropdown/Number), Risk Score (Calculated Field or Manual), Response Strategy, Response Plan Summary, Risk Owner, Status, Trigger Conditions, Contingency Plan Summary, Last Reviewed Date, Link to related Issues/CRs.

- ****Alternative:** Dedicated Confluence page using tables, dynamically updated or linked to Jira queries.**

- **Probability & Impact Matrix:** Defined matrix (like the example above) documented in Confluence as part of the Risk Management Plan.**
- **Risk Breakdown Structure (RBS - Optional):** A hierarchical decomposition of potential risk sources (e.g., Technical, External, Organizational, Project Management) used during identification workshops. Documented in Confluence.**
- **Meeting Minutes:** Weekly Status, MBR, Retrospectives capturing risk discussions and decisions (Confluence).**

- **Reports:** Risk sections integrated into Weekly Status Reports and MBR Presentations. Jira/ADO dashboards visualizing risk exposure (e.g., number of open risks by score/owner).

D. Roles & Responsibilities Summary (Risk Management)

- **All Team Members/Stakeholders: Responsible for identifying and raising potential risks.**
- **Project Manager: Owns and facilitates the overall risk management process, maintains the Risk Register, ensures risks are analyzed and have owners, reports on risk status.**
- **Risk Owner: Assigned individual responsible for developing, implementing, and monitoring the response plan for a specific risk.**
- **SMEs: Provide expert input during risk identification, analysis, and response planning.**
- **Scrum Master: Facilitates risk identification during Agile ceremonies.**

- **Steering Committee: Provides oversight, makes decisions on high-impact risks, accepts residual strategic risks.**

V. Integrating Formal PM with Agile/DevOps

These formal Change and Risk Management processes are designed to overlay, not replace, the Agile framework:

- ****Change vs. Backlog:** The CCB handles changes *to the baseline*. The Product Owner manages priority and refinement *within* the baseline via the backlog. Small adjustments or clarifications identified during grooming are typically *not* formal CRs. Significant deviations emerging from grooming *should* trigger a CR.**
- ****Risk Actions in Sprints:** Mitigation or contingency actions identified in the Risk Plan become concrete tasks or User Stories in the Product Backlog, prioritized by the PO alongside features. The work gets done within sprints.**
- ****Transparency:** Jira/ADO serves as the central hub. CRs and Risks are issue types alongside User Stories and**

Bugs, providing integrated visibility on project boards and dashboards.

- **Cadence Alignment:** CCB meetings and Risk reviews are scheduled to align with sprint cadences and MBR reporting cycles, ensuring timely information flow.
- **Focus:** Agile focuses on *delivering value iteratively within defined boundaries*. Formal PM focuses on *managing changes to those boundaries* and *proactively addressing threats* to achieving the overall objectives.

VI. Implementation How-To Summary

1. Configure Tools: Set up custom issue types, workflows, and fields for "Change Request" and "Risk" in Jira/ADO. Create associated dashboards and filters for Registers.
2. Develop Templates: Create Confluence page templates for the Change Management Plan, Risk Management Plan, CCB Charter, CCB Minutes, Impact Assessment, Risk Register details (if using Confluence), and meeting agendas.
3. Establish CCB: Formally define CCB membership, charter (scope, authority, decision process), meeting schedule,

and operating procedures.

4. **Conduct Initial Workshops: Run the project kick-off, initial requirements workshops, and the initial Risk Identification workshop.**
5. **Train the Team: Conduct dedicated training sessions for the entire project team and key stakeholders on the defined Change Management and Risk Management processes and associated tools (Jira, Confluence).**
6. **Integrate into Cadence: Incorporate CCB meetings and Risk Register reviews into the regular project meeting schedule defined in the Communications Plan.**
7. **Monitor & Refine: Continuously monitor the effectiveness of the processes via retrospectives and stakeholder feedback, making adjustments as needed.**

Responsibility: Project Manager, Jira/Confluence Administrator, Scrum Master, CCB Chair.