Phase 1 Systems Engineering Management Plan (SEMP)

ARC ITS4US Deployment Project

www.its.dot.gov/index.htm

Final Report — February 14, 2022
FHWA-JPO-21-915

COMPLETE TRIP
ITS4US

U.S. Department of Transportation

Produced by Atlanta Regional Commission (ARC)
U.S. Department of Transportation
Intelligent Transportation Systems (ITS) Joint Program Office (JPO)
Federal Highway Administration (FHWA)
Office of the Assistant Secretary for Research and Technology
Federal Transit Administration (FTA)

Notice

Technical Report Documentation Page

1. Report No. **FHWA-JPO-21-915**	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Phase 1 Systems Engineering Management Plan (SEMP) ARC ITS4US Deployment Project		**5. Report Date** February 14, 2022 **6. Performing Organization Code** 075863845
7. Author(s) Kofi Wakhisi (ARC), Maria Roell (ARC), Polly Okunieff (GO Systems and Solutions), Natalie Smusz-Mengelkoch (Kimley-Horn & Associates), JD Schneeberger (Kimley-Horn & Associates)		**8. Performing Organization Report No.** (Delete and insert information here or leave blank)
9. Performing Organization Name and Address Atlanta Regional Commission – Georgia Ste Gov Atlanta RGL COM 229 Peachtree St NE, Ste 100 Atlanta, GA 30303-1601		**10. Work Unit No. (TRAIS)** **11. Contract or Grant No.** 693JJ321C000008
12. Sponsoring Agency Name and Address U.S. Department of Transportation ITS Joint Program Office 1200 New Jersey Avenue, SE Washington, DC 20590		**13. Type of Report and Period Covered** Final **14. Sponsoring Agency Code** HOIT-1
15. Supplementary Notes Elina Zlotchenko (Program Manager), Amalia Rodezno (Contracting Officer), Karen Timpone (Contracting Officer Representative)		
16. Abstract The Atlanta Regional Commission Complete Trip - ITS4US Deployment project, Safe Trips in a Connected Transportation Network (ST-CTN), is leveraging innovative solutions, existing deployments, and collaboration to make a positive impact using transportation technology to support safety, mobility, sustainability, and accessibility. The ST-CTN concept is comprised of an integrated set of advanced transportation technology solutions (connected vehicle, transit signal priority, machine learning, predictive analytics) to support safe and complete trips, with a focus on accessibility for those with disabilities, aging adults, and those with limited English proficiency. This document serves as the Systems Engineering Management Plan (SEMP) for the deployment project. The SEMP provides an overview of the system engineering processes that will be implemented and defines the processes for design, development, integration, and testing. The SEMP will also outline the Agile development processes used in the deployment.		
17. Keywords ITS4US; Complete Trip; Deployment; ITS; Intelligent Transportation Systems; Systems Engineering Process; Systems Engineering Management Plan (SEMP)	**18. Distribution Statement**	

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
None	None	83	N/A

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

Revision History

Name	Date	Version	Summary of Changes	Approver
EMT / Tech Team	November 18, 2021	0.1	Initial Draft	JD Schneeberger
EMT / Subsystem Developers	November 29, 2021	0.2	Draft	Maria Roell
EMT	February 14, 2022	1.0	Final	Maria Roell

Table of Contents

List of Tables

List of Figures

1. Introduction

The Safe Trip in a Connected Transportation Network (ST-CTN) project is leveraging innovative solutions, existing deployments, and collaboration to make a positive impact using transportation technology to support safety, mobility, sustainability, and accessibility. The ST-CTN concept is comprised of an integrated set of advanced transportation technology solutions (connected vehicle (CV), transit signal priority (TSP), machine learning, predictive analytics) to support safe and complete trips, with a focus on accessibility for those with disabilities, aging adults, and those with limited English proficiency (LEP).

A well-defined systems engineering process is needed to support the planning, design, deployment, operations, and maintenance of the system that will be deployed as part of the ST-CTN project. The scope of this System Engineering Management Plan (SEMP) is to document the systems engineering process the project team will follow to deliver a successful ST-CTN project. Benefits of having a well-defined systems engineering process include:

- Improved stakeholder participation
- More adaptable, resilient, and interoperable systems
- Verified functionality and fewer defects
- Replicability and continuity with subsequent projects
- Better documentation

The SEMP will be a living document and used to guide activities in Phase 1, Phase 2, and Phase 3 of the project. As the initiative evolves and the team acts on the described activities and process, some elements may require refinement to ensure a quality and sustainable system is deployed. The SEMP will be updated as needed to accommodate these changes and refinements to ensure that the project team understands and continues to follow systems engineering processes that will result in successful program outcomes.

1.1 Document Purpose

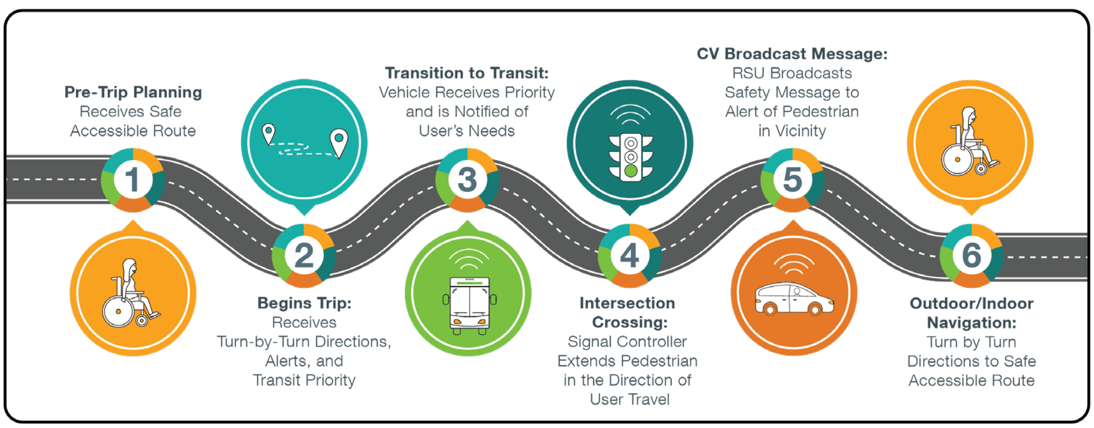
The purpose of the SEMP is to provide an overview of the systems engineering processes that will be used during the design, deployment, and test phase as well as the remainder of the project’s lifecycle. A rigorous application of established systems engineering processes will enable an efficient, interoperable, and replicable complete trip deployment while avoiding the risk of schedule delay or cost overruns. The SEMP builds on the User Needs Identification and Requirements Planning (UNIRP) document, created in Phase 1 of the project, to further define the systems engineering processes for design, development, integration, and testing. The SEMP defines how Vee-Processes and Agile Processes will be used to deliver the project.

1.2 Project Overview

The ST-CTN concept will integrate five programs currently existing or underway with regional commitments into a single system. These programs are leading the Atlanta region towards providing all travelers with a suite of innovative mobility solutions, which will be leveraged to support the ST-CTN system. ST-CTN will merge these separate initiatives through data fusion and communications network integration. Atlanta Regional Commission (ARC) intends to leverage the successes of the infrastructure, tools, and capabilities of these programs and apply them to support trip planning and wayfinding for all travelers, particularly underserved communities.

The goal of the ST-CTN system is to leverage existing advanced transportation technology solutions to support ***safe, reliable, accessible, complete*** *trips for all, particularly undeserved communities, including people with disabilities, aging adults, people with LEP, and low-income travelers.*

The ST-CTN project aims to upgrade and integrate existing technologies and services to assist underserved populations with completing their trip successfully, safely, and reliably. The vision of the project is to provide users complete trip functionality with directions, conditions, and status on the links between trip legs that are personalized based on the user's profile, while connecting the user to CV infrastructure to provide safer trips and more transportation network awareness. Transit based trips were delineated into 6 segments (as depicted in **Figure 1**) to allow for easier understanding and a greater breakdown of priorities and goals.



Source: ARC, 2020

Figure 1. Traveler's Complete Trip

The delineated trip segments include the following steps and project components:

- **Step 1 Pre-Trip Planning.** The traveler plans for and receives a safe accessible route.
    - The ability to customize trip preferences based on the user's abilities.

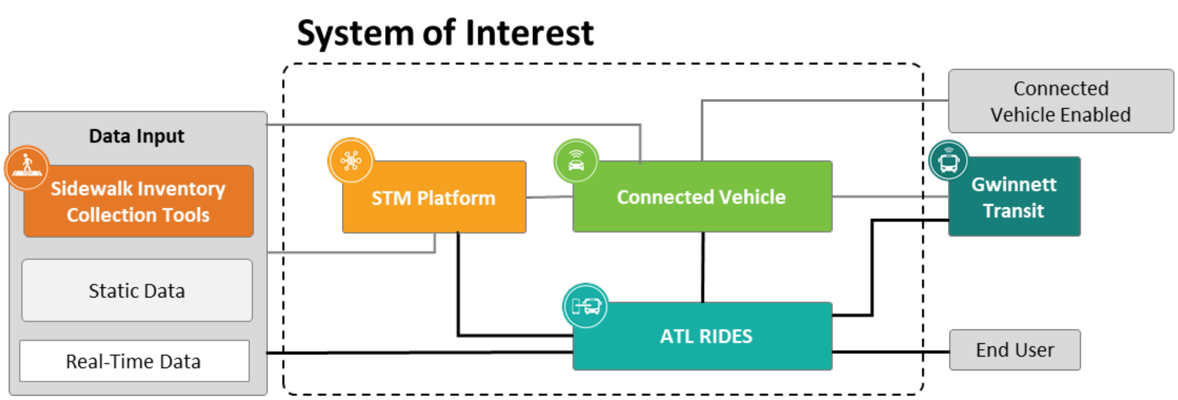
- **Step 2 Begins Trip.** The traveler begins their trip and receives turn-by-turn directions, alerts, remote pedestrian activation, and can trigger TSP if the user requires additional time boarding or alighting a transit vehicle, is unable to stand for long periods, or is sensitive to weather conditions.
  - Turn-by-turn, shortest path, directions along pathways that meet user defined preferences.
  - Provides support services for users if they become disoriented or have issues accessing defined paths.
  - Activates TSP for buses if the user requires additional time boarding or alighting a transit vehicle, is unable to stand for long periods, or is sensitive to weather conditions.
- **Step 3 Transition to Transit.** The traveler transitions to transit and the transit vehicle receives priority and is notified of users' needs. TSP can be triggered if the bus is running behind schedule due to a longer boarding time needed by a user.
  - Provides users with transit trips that have accommodations that meet user defined preferences.
  - Sends alerts to transit vehicles when users need additional time to board, navigate internally, or alight the transit vehicle.
  - Remotely requests service from transit vehicles while waiting to board or alight.
  - Triggers TSP if the bus is running behind schedule due to a user needing additional time to board or alight.
- **Step 4 Intersection Crossing.** When crossing a signalized intersection, the traveler interacts with the signal controller which extends the pedestrian phase in the direction of user travel.
  - Allows the user to communicate with connected intersections if they are unable to reach or press the crosswalk button.
  - Provides the user with information about the intersection crossing and adds time to the crossing if needed.
- **Step 5 CV Broadcast Message.** Roadside units (RSUs) broadcast safety message to alert CVs of pedestrians/bicyclists in the vicinity.
  - Provides the ability for users to remotely request service from transit vehicles while waiting to board or alight.
  - Provides communications between CVs and users to make them aware of each other when crossing a roadway or waiting at a transit stop.
- **Step 6 Outdoor/Indoor Navigation.** The traveler is provided with turn-by-turn directions to a safe accessible route.
  - Hands-free navigation via mobile apps and/or wearables and accessible channels (haptic, voice, text).
  - Alerts and dynamic rerouting in response to changes in path conditions.

- Provides the user with accessible routes into and through transit hubs within the project area.
- Provides users with updates on the operating status of indoor infrastructure such as elevators and escalators.

Additionally, user reporting will be available through the application to allow users to provide feedback on infrastructure that is currently out of service (elevators, escalators, etc.) or not accessible due to temporary or permanent obstructions (e.g., missing sidewalks or shared use-paths, etc.). This feature will help users avoid becoming delayed or stranded because of unforeseen outages. Transit providers, city, county, and/or construction crews currently flag outages into the system. New features being proposed as part of the project will allow users to flag infrastructure that has not already been flagged by public agency staff. System development and system integration completed within the scope of this pilot will enable travelers – specifically those in the underserved community – to program and safely complete single mode or multimodal trips that are based on their abilities; improve the transition between modes by providing additional details to users and transit service operators; suggest dynamic routing changes based on infrastructure condition and calculated delay; and use crowdsourced data collection to update infrastructure conditions.

The ST-CTN project will use open-source software (OSS) tools allowing for the results to be replicable across the region and sidewalk inventory innovations will reduce the costs of managing pedestrian assets in any community.

The conceptual diagram presented in **Figure 2** illustrates the concept, including the software, hardware, communications, and services planned for ST-CTN as shown.

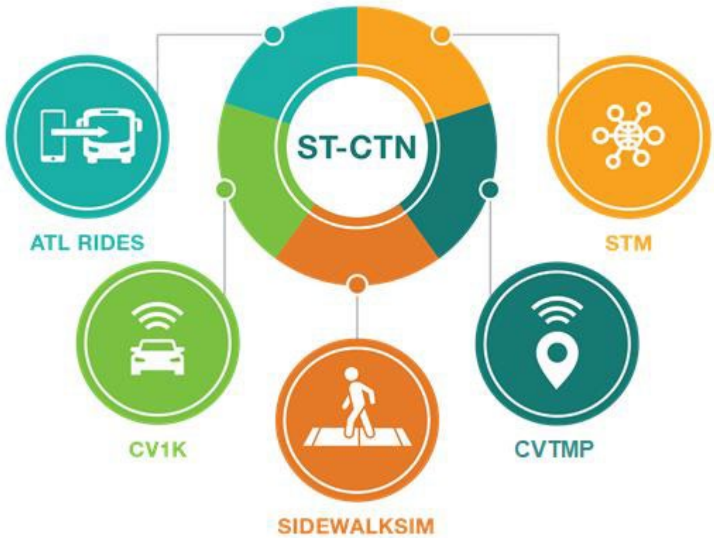


Source: ARC, 2021

Figure 2. ST-CTN High-Level Context Diagram

The scope of the project is limited to development of interfaces between existing programs that expand the capabilities of these programs. The existing initiatives that are being leveraged to support the proposed ST-CTN system are shown in **Figure 3** and defined in more detail below.

These icons and colors are used throughout the Phase 1 documents to clearly identify the critical components of ST-CTN.



Source: ARC, 2020

Figure 3. ST-CTN Integrated Initiatives

ATL RIDES. Atlanta-Region Ride Information and Data Evaluation System (ATL RIDES) includes an OSS multi-modal trip planning and mobile application, integrated mobile fare payment options, and a Connected Data Platform (CDP) using regional General Transit Feed Specification (GTFS). The tool supports multi-agency context, multilingual support, and live-tracking capabilities using GTFS feeds. The Open Trip Planner (OTP) architecture facilitates integration with additional OSS tools including a data analytics engine, call center with integrated voice response (IVR), and account management system.

SIDEWALKSIM. SidewalkSim is an asset management system and shortest path (lowest impedance) routing tool for pedestrian pathways. Site inspections provide more detailed Americans with Disability Act (ADA) and inclusive design and condition data for use in pathway accessibility analysis. SidewalkSim identifies the best path between any two points in the pedestrian network, given the set of pathway characteristics and any user-specified needs and route penalties.

CV1K. The Atlanta region is home to one of the largest CV deployments in the United States – Regional Connected Vehicle Infrastructure Deployment Program (CV1K). CV1K is deploying interoperable CV technologies at signalized intersections throughout the Atlanta region using both Dedicated Short-Range Communications (DSRC) and Cellular Vehicle to Everything (C-V2X) technologies to deliver safety and mobility-based applications. The program provides support to configure, operate, and maintain CV infrastructure and applications, including TSP. Gwinnett County will be one of the largest recipients of the first phase of this deployment.

CVTMP. Gwinnett County's Connected Vehicle Technology Master Plan (CVTMP) sets out to develop and improve economic viability and quality of life, address the needs and challenges to motorized and non-motorized modes, establish guidelines for deploying technology, and have broad applicability to Gwinnett, other local jurisdictions, and across the state—to set the standard for implementing CVs. Among the high priorities is establishing a mobile accessible safety program and alternative strategies for TSP in Gwinnett County.

STM. The Space Time Memory (STM) platform processes traffic volume and speed data from multiple monitoring and modeling sources, tracks network performance measures, and predicts evolving route conditions using traditional and machine learning techniques. The STM projects trip trajectories through the transportation network, as network conditions change in space and time. This tool will be applied to analyze and predict performance through the multi-modal transportation network. The shortest path analysis will be applied to the combined roadway, transit, sidewalk, and shared-use path networks, allowing routing decisions to incorporate travel time, safety, and other costs into path selection.

In some cases, partner agencies are upgrading the services within their current systems to create a more robust data set or toolset for the ST-CTN program; details on these upgrades and changes are discussed further in **Section 2**.

1.3 Organization of the Report

The SEMP includes the following sections, which detail the systems engineering processes required for a successful deployment of the ST-CTN project.

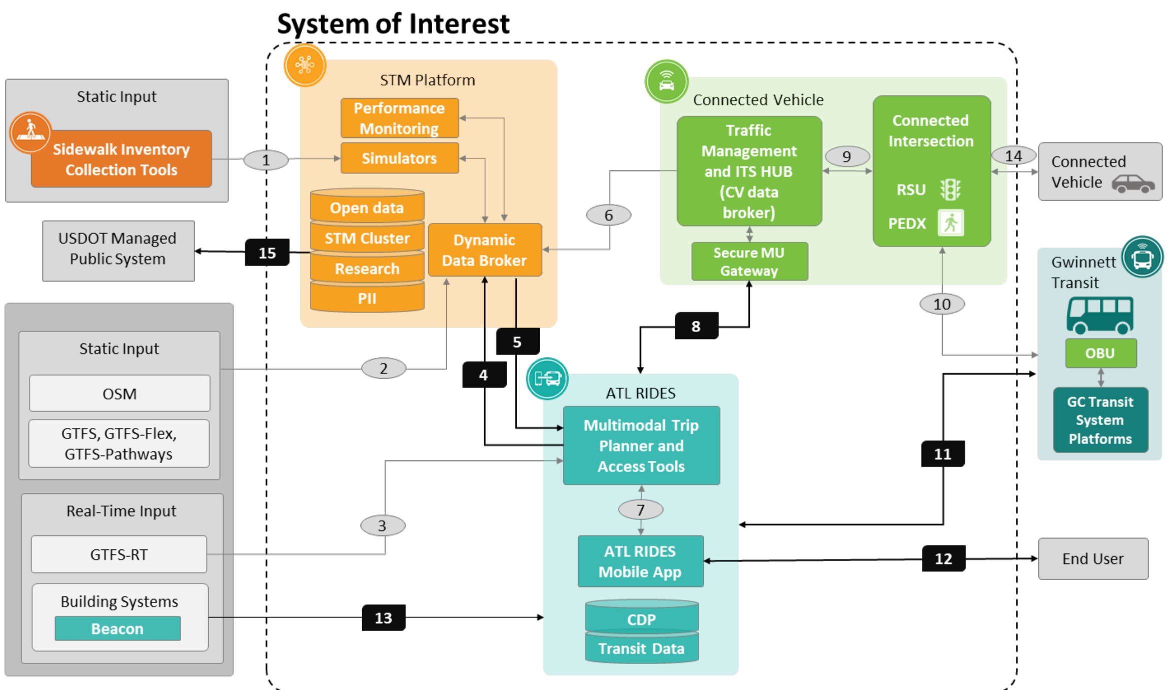
- **Section 2** (System Overview) provides an overview of the ST-CTN system. It includes relevant context diagrams and discusses the constraints of the system.
- **Section 3** (Systems Engineering Process Application) defines the systems engineering processes the project team will utilize in delivering the ST-CTN project. It includes approaches for systems engineering process planning as well as systems engineering technical processes. Finally, it includes processes relevant to Phase 3 and continued project operations and maintenance.
- **Appendix A** (Acronyms and Glossary) provides the acronyms and glossary definitions for technical terms used throughout this document.
- **Appendix B** (References) includes a list of documents referenced during the plan, including URLs and USDOT Publication Numbers, where possible.
- **Appendix C** (Source Code Management Plan) describes the multiple open-source code storage repositories, all hosted in GitHUB, that will be used to manage source code for the ST-CTN project.

1.4 SEMP Update Process

The ST-CTN SEMP is a living document that will be updated periodically over the life of the project when (or if) any SE processes have changed. If updates to the SEMP are required, the document will be updated and reviewed by the Executive Management Team (EMT) before transmittal to the USDOT. The SEMP is the primary planning document for the systems engineering technical elements. It defines systems engineering processes and methodologies used on the project and the relationship of SE activities to other project activities. The Program Management Plan (PMP) is the primary planning document for the overall project. It describes all activities, including technical activities, to be integrated and controlled during the life of the program. The Phase 2 and Phase 3 PMP will take precedence over any processes identified in this document. In addition, during Phase 3 and Post Phase 3, the Operations and Maintenance document will take precedence over this document. As such, the SEMP must be consistent with and evolve in concert with the PMP and Operations and Maintenance document.

2. System Overview

The five subsystems described in **Section 1.2** gave a broad overview of the system context. **Figure 4** provides more detail about how the subsystems will interact and be integrated under the ST-CTN system. The context diagram defines the internal and external subsystems as well as the data exchanges, interfaces to external systems, and interactions between the various components.



Source: ARC, 2021

Figure 4. Safe Trip in a Connected Transportation Network Data Exchange Flow Diagram

The ATL RIDES subsystem is contained within the light blue box in the lower-middle of the System of Interest. The ATL RIDES subsystem will contain the ST-CTN application and will be the subsystem that the end users interact with. The CV subsystem is contained within the green box in the upper-right of the System of Interest. The CV subsystem contains the components needed to receive and send CV messages and TSP or pedestrian crossing requests to connected signal controllers. The STM Platform subsystem is contained within the orange box in the upper left of the System of Interest. The STM Platform contains the processing system which evaluates routing choices and assigns impedance values. The Gwinnett County Transit (GCT) subsystem is contained within the teal box to the right of the System of Interest. The GCT subsystem contains on-board units (OBUs) and the GCT central system. Other external inputs include the static or real-time input of data, represented by the grey boxes on the left side of the System of Interest; vehicles with CV hardware, represented by the grey box on the upper-right side of the System of

Interest; and the end user, represented by the grey box on the lower-right side of the Systems of Interest.

Critical ST-CTN data exchanges are identified by number in the context diagram above and described in **Table 1**. The grey oval labels indicate existing data exchanges that will be utilized with no change to the current data exchange. Black rectangular labels indicate data exchanges that will be new or upgraded to support the ST-CTN system.

Table 1. Critical ST-CTN Connection Descriptions

Data Exchange ID (EX ID)	Description
1	Sidewalk inventory data, including accessibility features to the STM Platform simulators
2	Static and dynamic data from various existing sources to the STM Platform dynamic data broker
3	Static and dynamic data from various existing sources to the ATL RIDES multimodal trip planner and access tools
4	Mobile App logs, trip feedback and crowdsourced data (introduced in this systems requirement document as the Asset Condition application programming interface (API))
5	STM Network Impedance API
6	CV and Traffic Operations Messages: signal phasing and timing (SPaT), Map Data (MAP), CV Advanced Traveler Information System (ATIS) broadcast data, NaviGAtor intelligent transportation system (ITS), road characteristics, traffic data
7	Open Trip Planner APIs and ATL RIDES APIs
8	Mobile Accessible Pedestrian Signal System (PED-SIG) / Personal Safety Message (PSM)
9	CV messages
10	TSP and other CV application messages
11	GCT computer-aided dispatch (CAD) application transactions for transit applications including Transit Connection Protection (replacing CV Transit Stop Request (TSR) and Pedestrian Transit Indication (PTI) as described in ConOps Version 1.0)
12	ATL RIDES and Traveler exchange – profile, trip plan, settings, notifications, feedback, etc.

Data Exchange ID (EX ID)	Description
13	Static and dynamic information from building facilities, including beacons, to ATL RIDES
14	CV Data
15	Project data for USDOT-managed Public System

2.1 System Constraints

The following system constraints have been identified during the development of the proposed system and will guide the design, deployment, management, and operations of the system.

- Routing Preferences – Too many profile preferences will reduce the options for trip planning. Some restrictions on preference choices may need to be put in place to address the end user's major accommodations yet still provide options for travel. Even if end user preferences are included in the scenarios, the priority for implementing them will be driven by ensuring that the data to address the preferences can be economically collected and a complete trip plan can be generated. Note: During the Agile development phase, stakeholders will contribute to the types of preferences incorporated into the application to verify that the system is not overly constrained such that a traveler cannot receive a trip plan.
- IT Policies – The ST-CTN system will be constrained by agency IT policies. The system will be designed to abide by agency IT policies.
- Operations and Maintenance – The deployment of the ST-CTN system will be designed such that the system will not require additional agency staff to operate and maintain the system.
- FCC Rules – CV technologies shall adhere to the FCC rules for the use of the Safety Band.
- Georgia Department of Transportation (GDOT) Network Security – GDOT's network security policies forbid any devices directly connected to their signals from also connecting to the cloud. This constraint has been addressed through the use of a Secure Mobile Unit Gateway (SMUG) in the CV diagram.

Further information regarding system constraints may be found in the ConOps, Sections 4.4, 6.2, and 6.3.

3. Systems Engineering Process Application

This Section defines, at a high level, the systems engineering processes that the ST-CTN project team will utilize for the remainder of the project lifecycle. The ST-CTN project team structure, roles, and responsibilities that will be utilized for Phases 2 and 3 are defined. Key project deliverables, milestones, and decision gates are documented. Standardized processes and Agile development tools are identified. In addition, the systems engineering technical processes that will be implemented to guide the project development and deployment are explained. An Agile/Vee Hybrid methodology will be utilized by the ST-CTN project team where Agile (Scrum) will be combined with a traditional systems engineering process (Vee-Process). This methodology is clearly defined and explained.

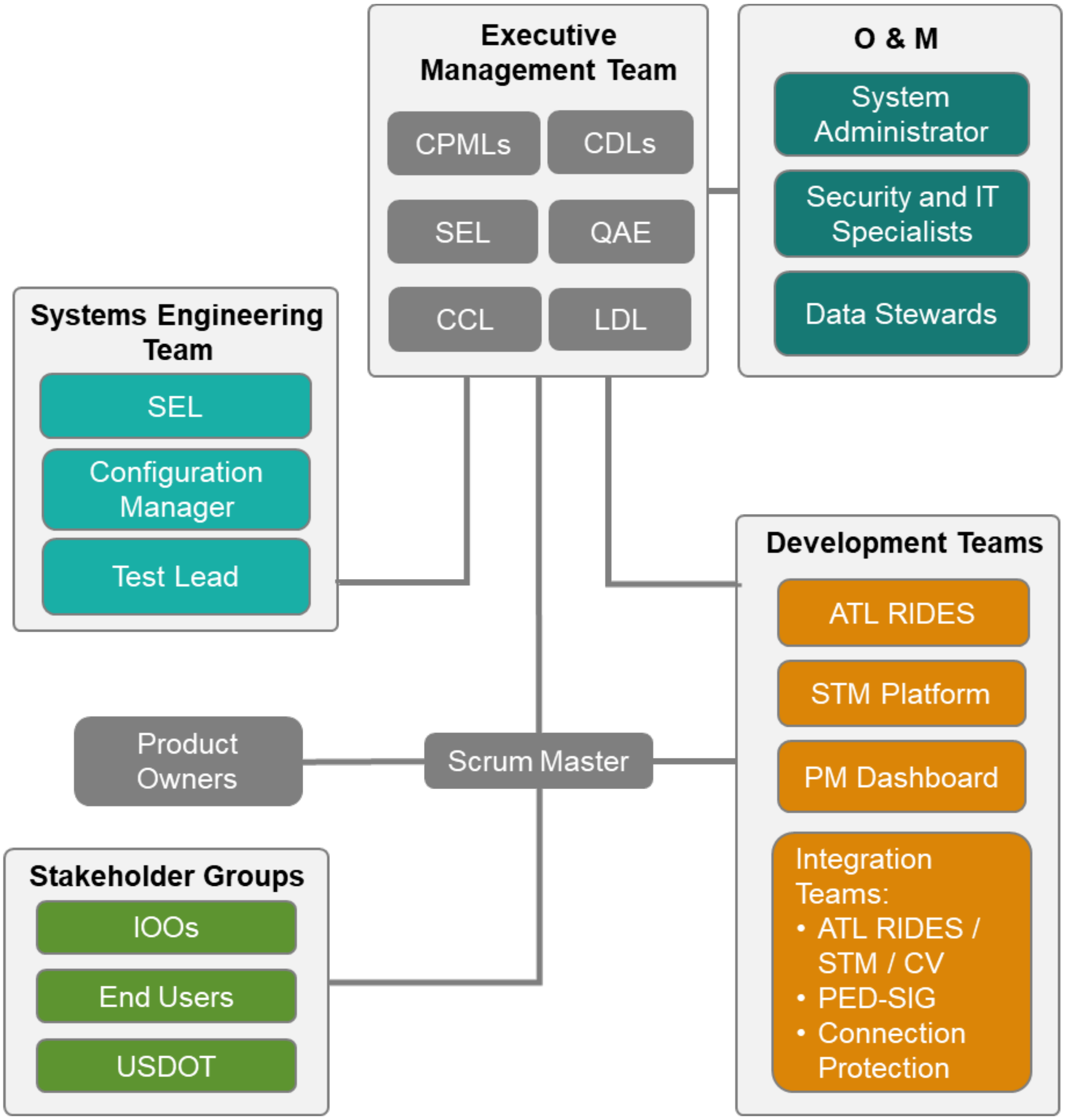
The SEMP is intended to be a living document and the following sections are populated with the best information available at the time of development, however, it is understood that the sections on later systems engineering processes are likely to be high level for now but will have additional detail added throughout the project lifecycle.

3.1 Systems Engineering Process Planning

This section contains information on systems engineering process planning for the ST-CTN project. It includes discussion on the project team organization, systems engineering deliverables, system milestones / decision gates, standardized processes, and Agile development tools that will be leveraged in delivering the project.

3.1.1 Project Team Organization

The Phase 2 Project team organization will be documented in the Phase 2 PMP. An excerpt of the ST-CTN project team is shown in **Figure 5**. The team is led by the ST-CTN Executive Management Team (EMT) who manages and controls schedule, resources, and project risks. The Systems Engineering Lead (SEL) will manage the technical development activities including the systems engineering and solutions effort, Agile development processes, and stakeholder integration into the design, development, testing, and review processes. The systems engineering roles and responsibilities of each entity in the organization chart are discussed in the sections below. Project administrative roles and responsibilities will be defined in the Phase 2 PMP.



Source: ARC, 2021

Figure 5. System Development Organization Chart

3.1.2 Executive Management Team

The EMT is comprised of the key personnel responsible for the successful delivery of the ST-CTN project, including:

- Co-Project Management Lead (CPML) – Alan Davis, GDOT
- Co-Project Management Lead (CPML) – Kofi Wakhisi, ARC
- Co-Deployment Lead (CDL) – Maria Roell, ARC
- Co-Deployment Lead (CDL) – Natalie Smusz-Mengelkoch, Kimley-Horn

- Systems Engineering Lead (SEL) – Polly Okunieff, GO Systems and Solutions
- Quality Assurance Engineer (QAE) – JD Schneeberger, Kimley-Horn
- Community Coordinator Lead (CCL) – Jordan Hall, SILC
- Local Deployment Lead (LDL) – Alex Hofelich, Gwinnett County Department of Transportation

The EMT will work together to manage ST-CTN project elements, with the CPMLs ultimately responsible for the successful completion of Phases 2 and 3 of the project. EMT responsibilities include:

- Collaborating with USDOT Agreement Officer (AO) and Agreement Officer Representative (AOR)
- Managing the ST-CTN project
- Guiding development of the ST-CTN concept
- Leading system development
- Leading coordination and engagement with Stakeholders, including end users
- Leading the development and production of systems engineering documents
- Leading the deployment and implementation of the concept
- Ensuring the safe, efficient, quality development, deployment, and implementation of the ST-CTN project

The EMT will meet regularly to manage and coordinate the project. The EMT will actively manage the ST-CTN project elements by working together to identify necessary work activities, assigning activities, developing interim milestone schedules, and following up to maintain an understanding of status and quality. Weekly EMT meetings will be conducted to monitor progress and implement mitigation strategies to issues or risks that arise. Based on the established team structure, members of the EMT will coordinate directly with specific leads and members within the ST-CTN project team.

3.1.2.1 Project Management Lead

The PML will be responsible for the quality and timely provision of project management artifacts required in the contract and for tracking project progress against target performance throughout the project lifecycle. The PML is responsible for risk tracking and risk mitigation. Duties include:

- Primary point of contact (POC) for the USDOT AO and AOR
- Ultimate responsibility for the quality and timely delivery of the ST-CTN project

3.1.2.2 Deployment Lead

The DL is responsible for managing, scheduling, controlling and monitoring the system development team. The DL will work with the EMT to coordinate all activities including training, outreach, performance measurement and marketing with the efforts undertaken by the System Development organization. Duties include:

- Meeting and coordinating system development tasks with technology teams
- Meeting and coordinating system development tasks between the development and other project teams
- Working with EMT to control resources and schedule
- Managing technical development activities
- Working with Product Owners to mitigate disruptions to their existing systems

The DL will meet regularly with technical staff to manage and coordinate the technical development activities. The DL will actively manage the ST-CTN system development project elements by working together with the EMT to identify and coordinate necessary work activities, assign activities, develop interim milestone schedules, and follow up to maintain an understanding of status and quality. Weekly meetings will be conducted between the DL and Development Teams to monitor progress and implement mitigation strategies to issues or risks that arise.

3.1.2.3 Deputy Project Manager

The DPM is responsible for supporting the PML and DL with their duties and responsibilities. The DPM will be responsible for coordination and management of the production of required deliverables. Duties include:

- Supporting the PML and DL with their duties and responsibilities
- Coordinating and managing required deliverable production
- Resource for ST-CTN project team members when additional support is required
- Mitigating identified project risks through resource management or additional collaboration
- Identifying key project areas needing additional support and coordinating additional support as required

3.1.2.4 Quality Assurance Engineer

The QAE is responsible for reviewing documents for quality and accuracy. The QA process is directed by the EMT prior to submission of each deliverable. The review duties include:

- Checking for conformance to document template
- Checking for content consistency within document and across document deliverables
- Checking for technical and editorial accuracy

3.1.2.5 Community Coordinator Lead

The CCL is responsible for engaging community groups and individuals. The CCL will have limited involvement or influence with the management of systems engineering for the ST-CTN project. However, coordination with the CCL on key project milestones and activities will be critical to ensure stakeholders are receiving accurate information and expectations.

3.1.2.6 Local Deployment Lead

The LDL is responsible for managing local deployment activities. The LDL will coordinate on-site activities such as testing and demonstrations. In addition, the LDL will monitor and report any local risks that may impact the deployment of the ST-CTN system.

3.1.2.7 Systems Engineering Lead

The SEL provides technical support and guidance to systems engineering and software development teams on processes, integration, and technology design, development, integration, and deployment strategies. As SEL, the staff person will work with the Configuration Manager and Test Lead to ensure that the systems engineering process is tracked and enforced. The SEL will also work with Agile teams to define the system architectures as well as ensure conformance to the Service Packages described in the Architecture Reference for Cooperative & Intelligent Transportation (ARC-IT). The SEL will be responsible for management of the Systems Engineering Team.

3.1.3 Systems Engineering Team

The Systems Engineering Team is responsible for ensuring that the systems engineering process is accomplished, tracing needs through each stage, and verifying and validating that the needs are met through the design, testing, and deployment phases. As described above, the SEL is responsible for leading the Systems Engineering Team.

3.1.3.1 Configuration Manager

Reporting to the SEL, Poonam Patel will serve as the Configuration Manager and will track the change management requests as the system needs and requirements continue to be refined through the design, coding, and testing phases.

The Configuration Manager will track all deliverables during Phases 2 and 3 and will be responsible for performing the following duties:

- Managing system engineering changes during stakeholder meetings, Sprint development, and testing
- Tracking system level change requests, particularly as they impact user needs and requirements, as the system continues to be refined through the design, coding, and testing processes
- Reviewing and verifying traceability for new system engineering artifacts
- Updating system engineering documents on an annual basis to ensure consistency

3.1.3.2 Test Lead

Joseph Yawn will serve as the Test Lead and will manage test cases and procedures to ensure that they are well formed, consistent, and that they validate and verify system requirements and needs. The Test Lead will manage and oversee all test planning including developing test procedures, cases, data sets, acceptance criteria, as well as manage test teams, demonstrations, and result reporting during Phases 2 and 3. Duties include:

- Setting standards for test planning during Agile development to ensure consistency throughout the process
- Reviewing integration and operational readiness test planning including but not limited to test cases, data, acceptance criteria, and results to ensure that test cases trace to user needs and requirements
- Working with the Scrum Master and Agile Development Teams to identify exit criteria. Pass/fail criteria and criticality level if it did not pass
- Tracks test cases to ensure testing validates system needs and verifies system requirements
- Facilitates periodic meetings with subsystem testers to coordinate integration testing and review testing efforts

The testers for each development team will meet periodically to review the testing approach and coordinate integration and operational readiness testing efforts under the leadership of the Test Lead.

3.1.4 Scrum Master

Mirwais Mojaddidi will serve as the Scrum Master and will manage the Agile development process (discussed in Section 3.2), including scheduling Sprints, facilitating Sprint Planning meetings, reviewing metrics, and most importantly, engaging stakeholders in prioritizing and refining user stories and acceptance criteria. The Scrum Master is accountable for the development team's effectiveness. Scrum Master duties include:

- Coaching the team members in self-management and cross-functionality
- Facilitating Product Backlog meetings with stakeholder groups including scheduling meetings, preparing agenda, presenting metrics, presenting backlog (including bug fixes), facilitating discussion with stakeholders, and documenting decisions and actions
- Facilitating Sprint Planning meetings with development teams to identify workload for next Sprint, identifying impacts to existing software (need for refactoring), reviewing Sprint development templates (see **Table 5**), and documenting decisions and actions
- Managing check-in meetings to assess progress during a Sprint
- Facilitating Sprint Reviews and demonstrations for Product Owners and stakeholder groups
- Hosting Sprint Retrospective meetings
- Supporting the removal of impediments to the team's progress

- Ensuring that all scrum events take place and are positive, productive, and kept within the timebox

3.1.5 Development Teams

A separate development team will be formed for each ST-CTN system subsystem and key development component. In addition, integration teams will be set up to handle system interfaces between subsystems and multi-subsystem applications. The integration teams will be set up on an ad hoc basis when Sprints include related user stories (e.g., PED-SIG). The development teams for stand-alone systems (prior to integration with other subsystems or external systems) include:

- ATL RIDES – IBI will lead the development of the mobile application and website modules which will require development to support ST-CTN system function and features
- STM Platform – GA Tech (Dr. Guensler and Dr. Guin) will lead the development of processing, network, and impedance calculations which will require development and enhancement to integrate and support ST-CTN system requirements
- Performance Measurement Dashboard – GA Tech (Dr. Guensler and Dr. Guin) will lead the development of the performance measurement dashboard which will provide the open data portal for accessing key performance measurement and evaluation metrics
- Integration Teams – Integration teams will be ad hoc. These teams will be cross-cutting and will include a variety of roles (e.g., design, develop, test, SE) to perform integration of subsystems and connected vehicle components, including:
    - ATL RIDES / STM / CV
    - PED-SIG
    - Connection Protection

Developers are responsible for implementing the Agile process and are committed to accomplishing the development goals identified during each Sprint Planning meeting. The specific skills needed by the developers are broad and will vary with the domain of work. However, developers are accountable for:

- Designing, implementing, reviewing and/or testing user stories that meet specific acceptance criteria including error processing
- Instilling quality by adhering to a Definition of Done
- Adapting their plan each day toward the Sprint Goal
- Completing and implementing the software development lifecycle (SDLC) process (implemented by **Table 5**).
- Following Agile processes including development, review, test, and deploy (illustrated in **Figure 7**)

3.1.6 Product Owner

The Product Owner for the ST-CTN system is Maria Roell. The Product Owner is a member of the Agile Team responsible for working with the team to define User Stories and prioritize the Backlog to streamline the execution of program priorities while maintaining the conceptual and technical integrity of the features or components for the team.

The Product Owner is accountable for maximizing the value of the product resulting from the work of the Development Teams. How this is done may vary widely across organizations, Development Teams, and individuals. The Product Owner is also accountable for effective Product Backlog management, which includes:

- Developing and explicitly communicating the Product Goal
- Creating and clearly communicating Product Backlog items
- Prioritizing and scheduling Product Backlog items
- Ensuring that the Product Backlog is transparent, visible, and understood

The Product Owner will be supported by Project Sponsors, including the USDOT AOR, Gwinnett County Department of Transportation (GCDOT), and GCT. In particular, GCT and GCDOT will actively participate in the Agile epics related to bus and intersection interface components.

3.1.7 Stakeholder Groups

The stakeholder groups consist of end users, infrastructure owners/operators (IOO), USDOT, and, as needed, communities of practice. Their duties include:

- Participating in stakeholder group meetings including Product Backlog Planning, Sprint Review and Demonstrations, and Sprint Retrospectives
- As appropriate, participating in alpha and beta testing system prior to release

3.1.7.1 Infrastructure Owners and Operators

IOO groups consist of representatives from buildings, right of ways, and other facilities for which data is acquired (collected or provisioned). IOOs may include owners or managers of public transit stations, facilities mapped for indoor navigation, parking lots, sidewalks, etc. who are responsible for repair and maintenance. Most of the stakeholders in this group are public sector staff, however, private sector participation may be recruited.

3.1.7.2 End Users and Communities of Practice

The end user stakeholder group consists of groups representing underserved communities and training coordinators. Other communities of practice, including the greater research community, will also be invited to participate as observers to advise on technology approaches, standards, or user experience (UX).

3.1.7.3 USDOT

USDOT, the project sponsor, and their representatives will be invited to participate in all stakeholder and Agile development team meetings. As project sponsors, USDOT will review all required deliverables for completeness, quality, and accuracy.

3.1.8 Systems Engineering Deliverables

Several systems engineering deliverables will be produced for the ST-CTN project. **Table 2** includes a list of Phase 1, Phase 2, and Phase 3 deliverables along with due dates and discussion on how each document will be maintained and configured. Phase 2 and Phase 3 systems engineering deliverable dates are subject to change based on the Notice of Funding Opportunity (NOFO).

Table 2. Systems Engineering Deliverables

Deliverable	Date	Maintenance	Configuration Control	Phase
Program Management Plan (PMP)	4/5/2021	Updated throughout phase; risk register, changes and schedule submitted monthly in MPR.	Original deliverable red-lined when changes to the document are identified.	1
User Needs Identification and Requirements Planning (UNIRP)	4/12/2021	Updated in SEMP.	Not updated. Will be updated in SEMP.	1
Needs Summary	5/3/2021	Updated in ConOps.	Not updated in Needs summary. Updated in ConOps.	1
Concept of Operations (ConOps)	6/28/2021	Reissued at the beginning of each phase. Phase 3 will include a final ConOps.	Original deliverable red-lined when changes to the document are identified.	1
Data Management Plan (DMP)	8/23/2021	Reissued at the beginning of each phase. The final will be issued at the end of Phase 3.	Original deliverable red-lined when changes to the document are identified.	1

Deliverable	Date	Maintenance	Configuration Control	Phase
Safety Management Plan (SMP)	8/23/2021	Reissued at the beginning of Phase 2.	Original deliverable red-lined when changes to the document are identified.	1
System Requirements Specifications (SyRS)	10/25/2021	Reissued at the beginning of each phase. The final will be issued at the end of Phase 3.	Original deliverable red-lined when changes to the document are identified.	1
Enabling Technology Readiness Assessment (ETRA)	11/15/2021	Reissued at the beginning of each Phase 2.	Original deliverable red-lined when changes to the document are identified.	1
Systems Engineering Management Plan (SEMP)	12/27/2021	Reissued at the beginning of each phase. The final will be issued at the end of Phase 3. Note: The SEMP will be assessed quarterly. If updates are required, the document will be updated semi-annually.	Original deliverable red-lined when changes to the document are identified.	1
Integrated Complete Trip Deployment Plan (ICTDP)	3/11/2022	Updated document will become the statement of work for phase 2.	Not applicable.	1
Deployment Readiness Summary Briefing (DRSB)	TBD	Not applicable.	Not applicable.	1

Deliverable	Date	Maintenance	Configuration Control	Phase
Program Management Plan (PMP)	3/29/2022	Updated throughout phase; risk register, changes and schedule submitted monthly in MPR.	Original deliverable red-lined when changes to the document are identified.	2
Data Privacy Plan (DPP)	6/21/2022	Reissued at the beginning of Phase 3 as needed.	Original deliverable red-lined when changes to the document are identified.	2
Software Development Schedule (SDS)	6/21/2022	Updated and issued every Sprint. This is envisioned to be the product roadmap and reflect the Sprint Metrics and Sprint Roadmap Workplan.	The SDS is anticipated to be an on-line tool or dashboard that shows the roadmap. The Sprint Planning Tool will drive the input to this dashboard. In addition, a baseline schedule, part of the PMP will be submitted showing the updates vs. the baseline schedule.	2
Systems Architecture Document (SAD)	8/16/2022	The final SAD will be issued at the end of Phase 3.	Original deliverable red-lined when changes to the document are identified.	2
Updated Phase 1 documents: ConOps Systems Requirements	8/16/2022	The final ConOps and SyRS will be issued at the end of Phase 3.	Updated deliverables red-lined when changes to the document are identified.	2

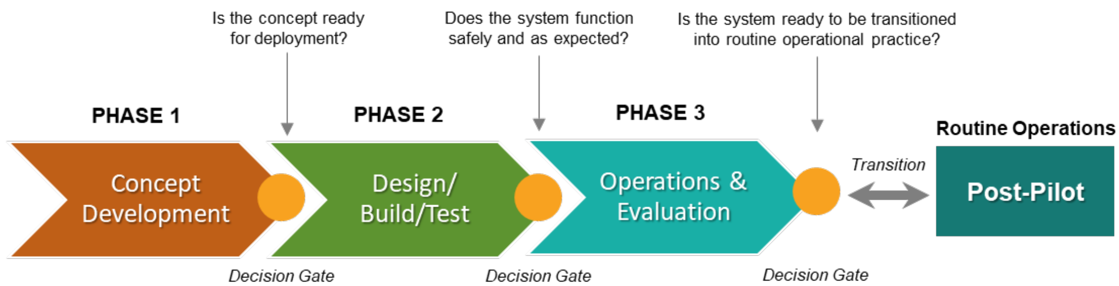
Deliverable	Date	Maintenance	Configuration Control	Phase
Data Management Plan (DMP)	9/13/2022	The final will be issued at the end of Phase 3. However, the Data Custodian Plan for each dataset will precede the DMP.	Updated Phase 2 deliverable red-lined when changes to the document are identified.	2
Comprehensive Acquisition Plan (CAP)	12/6/2022	Not applicable	Not applicable	2
Systems Design Document (SDD)	1/31/2023	The SDD is documented by user stories, module logic, Sprint code and traceability to needs and requirements. When design is refactored, information is documented to show change to the original design.	The SDD is anticipated to be an on-line description of the implemented Sprint designs. The tool managing the configuration will manage versions, authors, changes and dates of change.	2
Comprehensive Installation Plan	2/28/2023	Not applicable	Not applicable	2
System Test Plan (STP)	2/28/2023	The STP is documented by user story acceptance criteria, test procedures, test data, test cases and results traceability to needs and requirements.	The STP is anticipated to be an on-line description of the implemented Sprint review, test and integration tests. The tool managing the configuration will manage versions, authors, changes and dates of change.	2

Deliverable	Date	Maintenance	Configuration Control	Phase
Operational Readiness Plan (ORP)	2/28/2023	A draft and final version of the ORP will be issued per SOW.	Original deliverable red-lined when changes to the document are identified.	2
Installation and Operational Readiness Schedule (IORS)	2/28/2023	The schedule will be maintained as part of the project management effort.	The ORS will be managed by the project scheduling tools.	2
Comprehensive Maintenance and Operations Plan (CMOP)	6/20/2023	The CMOP is seen as a living document, particularly during Phase 3, when system reliability and operations are reviewed and exercised.	Original deliverable red-lined when changes to the document are identified.	2
System Operations and Maintenance Schedule (SOMS)	3/29/2024	The schedule will be maintained as part of the project management effort.	The SOMS will be managed by the project scheduling tools.	3
Outreach Implementation Schedule (OIS)	3/29/2024	The schedule will be maintained as part of the project management effort.	The OIS will be managed by the project scheduling tools.	3
Program Management Plan (PMP)	4/12/2024	Updated throughout phase; risk register, changes and schedule submitted monthly in MPR.	Original deliverable red-lined when changes to the document are identified.	3

Deliverable	Date	Maintenance	Configuration Control	Phase
Comprehensive Transition Plan (CTP)	3/28/2025	Final version will be submitted at the end of Phase 3	Original deliverable red-lined when changes to the document are identified.	3
SDO-Specific Technical Memoranda	Per Standards Plan within SAD	Not applicable.	Not applicable.	3

3.1.9 System Milestones/Decision Gates

This section includes discussion of the activities encompassing Phases 1, 2, and 3 of the ST-CTN initiative and how the project transitions between phases. The ST-CTN project is being delivered as part of the USDOT's Complete Trip - ITS4US Deployment Program which is being executed in three phases – Phase 1: Concept Development; Phase 2: Design, Build, and Testing; and Phase 3: Operations and Evaluation. **Figure 6** depicts the phases and decision gates for the initiative.



Source: ARC, 2021

Figure 6. ITS4US Complete Trip Program Phases and Decision Gates

Phase 1: Concept Development includes concept development for the ST-CTN project. In this phase, the preliminary proposed idea is developed into a structured concept that is suitable for further design, development, testing, and operation. Through stakeholder engagement activities, a set of documents are created to plan for and develop the concept for the project. These foundational documents guide the remainder of the initiative. Key Phase 1 systems engineering documents include:

- Concept of Operations (ConOps)
- Data Management Plan (DMP)
- Safety Management Plan (SMP)
- System Requirements Specification (SyRS)
- Enabling Technologies Readiness Assessment (ETRA)

- Systems Engineering Management Plan (SEMP)
- Integrated Complete Trip Deployment Plan (ICTDP)

Phase 1 culminates with the ICTDP which serves as a decision gate for going into Phase 2. The ICTDP is intended to serve two purposes: first, to summarize the refined deployment concept developed in Phase 1 activity and second, to set forth a high-level Phase 2 (Design/Build/Test) and Phase 3 (Operations and Evaluation) schedule. A refinement of the original Phase 1 proposal's high-level cost estimate for Phase 2 and Phase 3 is identified reflecting the refined deployment concept, the deployment schedule, and the more detailed examination of risks and requirements associated with at-scale deployment (technical, institutional, and financial) revealed during Phase 1 activity.

Phase 2: Design, Build, and Test includes activities to design, build, test, and deploy the ST-CTN project. This phase includes activities to design, build/implement, and test the ST-CTN system prior to the system becoming operational. Key Phase 2 systems engineering documents include:

- High Level Design Processes will include the following deliverables:
  - System Architecture Document (SAD)
  - Preliminary System Design Document (SDD)
  - Data Management Plan (Phase 2)
  - Data Privacy Plan
- Agile Processes will include the following artifacts and deliverables:
  - Iterative design, code, and test documentation
  - Comprehensive Acquisition Plan (CAP) (if applicable)
  - Comprehensive Installation Plan (if applicable)
  - Software Development-related Documentation
- System Verification and Deployment Processes will include the following deliverables:
  - Draft and Final SDD
  - System Test Plan (STP)
  - System Test Results
- System Validation Processes will include the following deliverables:
  - Operational Readiness Plan (ORP)
  - Operational Readiness Demonstrations
- Operations and Maintenance Processes will include the following deliverables:
  - Comprehensive Maintenance and Operations Plan (CMOP)

During Phase 2, development will be conducted and unit and system testing will occur to ensure that the system developed meets the needs and requirements established in Phase 1. An Agile Process will be used to incrementally develop the ST-CTN solution. Design, implementation, and testing will occur as part of the Sprint process. Testing will be done incrementally and summarized in the STP that will articulate how all of the requirements defined in the SyRS are met.

Phase 2 culminates with Operational Readiness Tests (ORTs) and Demonstration to assess that the system functions as required. Operational readiness is established with a comprehensive set of tests and supporting demonstrations to be designed and conducted by the ST-CTN project team. The ST-CTN project team will conduct a set of relevant tests to verify that the system performs according to the documented System Requirements. Demonstrations will also be conducted (at a higher level) and show that the system performs as expected in key use cases/scenarios. Relevant testing defined in the STP will be conducted *prior* to conducting the ORT and Operational Readiness Demonstration.

Details about the development of the system engineering documents in coordination with the Agile development process are described in **Section 3.2**.

Phase 3: Operations and Evaluation includes activities related to the operations and evaluation of the system. The system will be operational for a minimum duration of 18 months. During this time, the ST-CTN project team will operate and maintain the system. Key systems engineering deliverables and activities for phase 3 include:

- System Operations and Maintenance Schedule (SOMS)
- Operational Capability Showcase (and Plan)
- Public-facing Data
- Comprehensive Transition Plan (CTP)

After the three phases of the effort are complete, successful elements of the integrated complete trip deployment are expected to transition to become elements of routine operational practice. The region will be responsible for the funding of continued operations of successful elements of the deployments for a minimum of 5 years beyond the period of performance of the federal Complete Trip-ITS4US Deployment Program. As part of Phase 1, the ST-CTN team prepared an Institutional, Partnership and Financial Plan (IPFP) intended to codify and provide definitive documentation of stakeholder agreement on concept, objectives, institutional and financial arrangements necessary for the successful implementation and operation of the deployment.

3.1.10 Standardized Processes

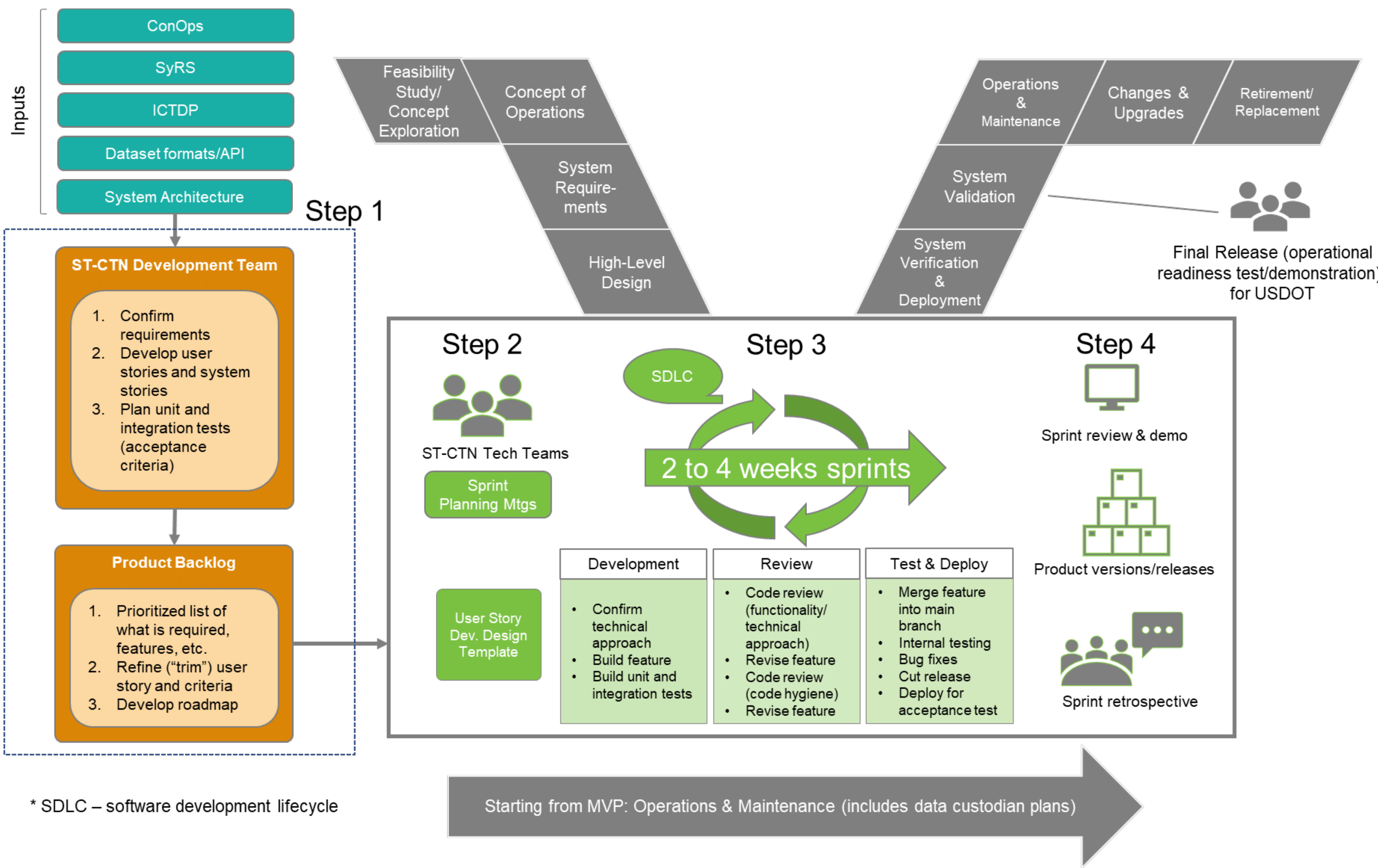
The systems engineering processes being used for delivering the ST-CTN project are based on standardized development processes including:

- International Council on Systems Engineering (INCOSE) Systems Engineering Handbook, Fourth Edition
- Systems Engineering Body of Knowledge (SEBOK)

- FHWA Systems Engineering for Intelligent Transportation Systems: http://www.ops.fhwa.dot.gov/publications/seitsguide/seguide.pdf
- International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC)/Institute of Electrical and Electronics Engineer (IEEE) 15288 Systems and software engineering – System lifecycle processes: http://www.iso.org/iso/catalogue_detail?csnumber=43564
- IEEE Standards
    - IEEE 1362-1998: IEEE Guide for Information Technology System Definition—Concept of Operations (ConOps) Document
    - IEEE 1233-1998: IEEE Guide for Developing System Requirements Specifications
    - IEEE 42010-2011: IEEE Recommended Practice for Software Architecture Descriptions
    - IEEE 1016-2009: Recommended Practice for Software Design Descriptions
    - IEEE Standard 1028-2008: Standard for Software Reviews and Audits
- The Scrum Guide™: https://scrumguides.org/scrum-guide.html
- Scrum Alliance: https://www.scrumalliance.org
- FHWA Applying Scrum Methods to ITS Projects: https://rosap.ntl.bts.gov/view/dot/32681

3.2 Systems Engineering Technical Processes

Delivery of the ST-CTN project will follow an Agile/Vee Hybrid methodology where Agile (Scrum) will be combined with a traditional systems engineering process (Vee-Process). The approach, depicted in **Figure 7,** establishes user needs, requirements, architecture, and conceptual design early in the process consistent with the Vee-Process and then pivots to an Agile process. The traditional stages of the Vee-Process will be followed until the system requirements and SAD are completed. These foundational components trace the needs and requirements to aid in the system deployment, while setting the stage for the Scrum process that allows for iterative and incremental development during the design and implementation stage. User needs and requirements (once developed and refined) will serve as inputs to epics and user stories in the Product Backlog. The Product Backlog will define the epics and user stories and establish the minimal viable product (MVP).



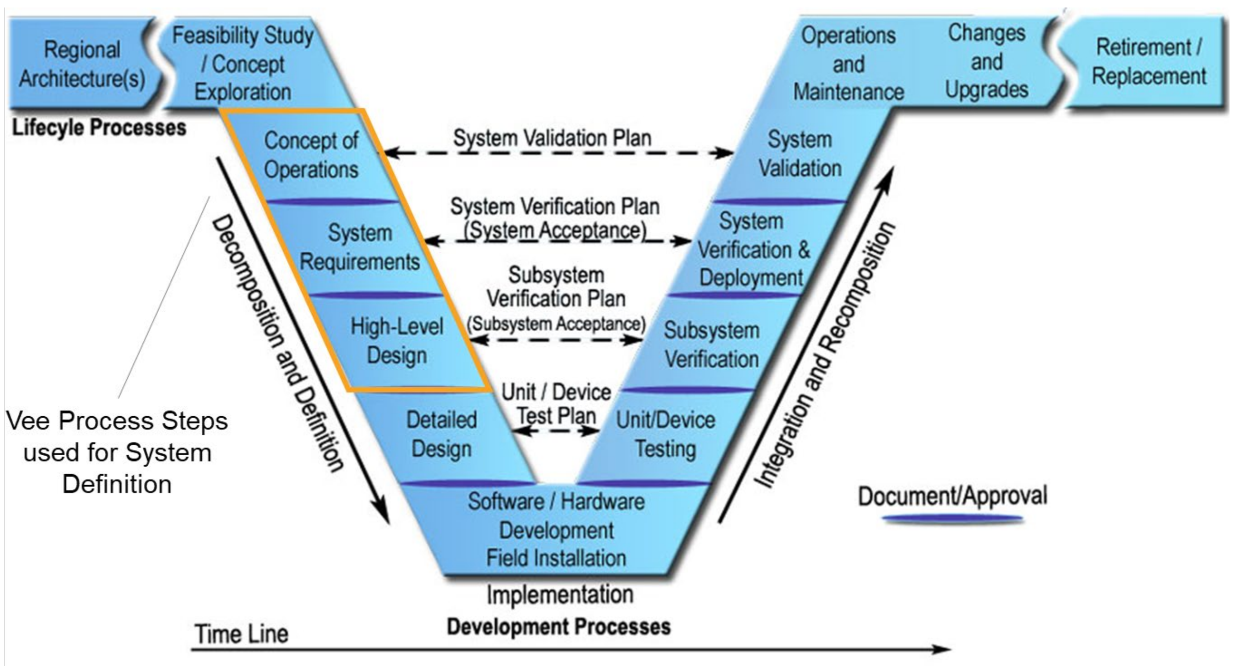
Source: ARC, 2021

Figure 7. The ST-CTN Agile/Vee Hybrid Methodology

Development of STN-CTN projects will then be incrementally delivered and tested during 2-to-4 week Sprints. Released Scrum software products will go through multiple Sprint cycles with each Sprint cycle delivering a portion of the product until the complete product (or planned increment release) is ready for integration with other parts of the system. When Scrum is complete, the process then reenters the Vee-Process for the system verification and validation (V&V) activities (in this case after merging into main branch and integration testing).

Vee-Process for System Definition

Delivery of the ST-CTN project will begin by following the Vee-Process for early and comprehensive identification of goals, a ConOps that describes user needs and the operating environment, thorough and testable system requirements, and defined system architecture, and SDD. As depicted in **Figure 8**, system definition of the ST-CTN project begins at the ConOps and ends at the SDD.



Source: FHWA, 2005

Figure 8. The Vee-Process

Key deliverables produced during the Vee-Process include:

- **Concept of Operations (ConOps).** The ConOps is a foundation document that frames the overall system and sets the technical course for the project. Its purpose is to clearly convey a high-level view of the system to be developed that each stakeholder can understand. The ConOps defines the scope of the project, the current situation, user needs, concept for the proposed system, operational scenarios, and summary of the impacts. The ST-CTN team developed the ConOps for the project during Phase 1 of the ITS4US Complete Trip Program. The ConOps was based on the USDOT's template which is based on IEEE 1362-1998: IEEE Guide for Information Technology System Definition—Concept of Operations (ConOps) Document.

- **System Requirements Specification (SyRS).** In the SyRS, stakeholder needs identified in the ConOps are reviewed, analyzed, and transformed into verifiable requirements that define *what* the system will do but not *how* the system will do it. Working closely with stakeholders, the requirements are elicited, analyzed, validated, documented, and baselined. The ST-CTN project team developed the SyRS during Phase 1 of the Complete Trip - ITS4US Deployment Program. The SyRS was based on USDOT's template which is based on IEEE 1233-1998: IEEE Guide for Developing System Requirements Specifications.

- **System Architecture Document (SAD) and Standards Plan.** The SAD describes the architecture for systems associated with the deployment and associated standards that will be used. The SAD includes the following architectures:

    - Enterprise Architecture. Describes the organizational structure and their relationships required to support the overall system architecture. In addition, the enterprise architecture describes the system drivers that impact the system.

    - Functional Architecture. Describes abstract functional elements (processes) and their information flows (web services and standards) that satisfy the system requirements.

    - Physical Architecture. Describes physical objects (systems and devices) and their application objects as well as the high-level interfaces between those physical objects. The physical architecture will use the ARC-IT Version 9 to describe the framework and will include communications components such as firewalls and routers to describe the communications.

  The SAD will be developed during Phase 2 of the Complete Trip - ITS4US Deployment Program. IEEE 42010-2011: IEEE Recommended Practice for Software Architecture Descriptions will be used as a guideline for format and content. The SAD also includes a Standards Plan that identifies the nature of required interfaces to other systems, which should be defined to utilize existing networking or other standards when available.

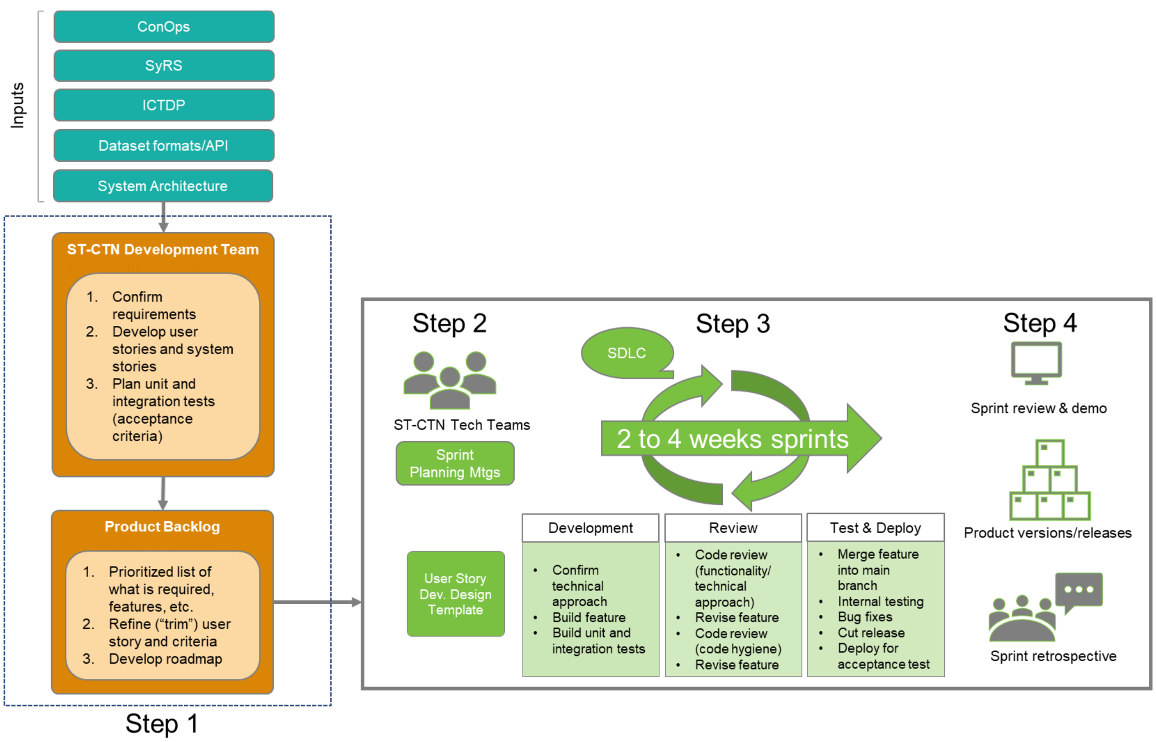
- **System Design Document (SDD).** The SDD includes a high-level design which facilitates development, integration, and testing of the ST-CTN system. The SDD, leveraging the Agile development methodology will describe the initial Product Backlog created based on the system requirements. System subsystem requirements are identified and allocated to Backlog Epics. These epics are further visualized as high-level user and system stories assigned to subsystem components and interfaces. Since the ST-CTN utilizes or leverages existing systems, the design document will reference existing design and interface specifications in the roadmap including dependencies. The ST-CTN project team will develop the high-level preliminary SDD at the onset of Phase 2. Design will continue throughout the Scrum Process and an as-built Draft and Final SDD will then be delivered at the end of the project. IEEE 1016-2009: Recommended Practice for Software Design Descriptions will be used as a guideline for format and content to develop the SDD.

Agile (Scrum) Process for Development, Implementation, and Testing

Once the Vee-Process for System Definition is completed, an Agile (Scrum) Process will be followed. Agile software development comprises various approaches to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their end users. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change.

Scrum is an Agile methodology in which the basic unit of development is called a Sprint. Every Sprint begins with a Sprint Planning meeting in which tasks for the Sprint are defined. This is also the part where the estimated commitment for the Sprint Goal is made. The Agile Process involves a team working through a complete Software Development Life Cycle (SDLC) process. Products will be developed incrementally through several iterations throughout a Sprint process. A Sprint concludes with a Sprint Review meeting and a Sprint Retrospective where progress is evaluated, and the lessons learned are shared.

An overview of the Agile Process is depicted below (shown as an extract from **Figure 7**) and defined by the following steps.



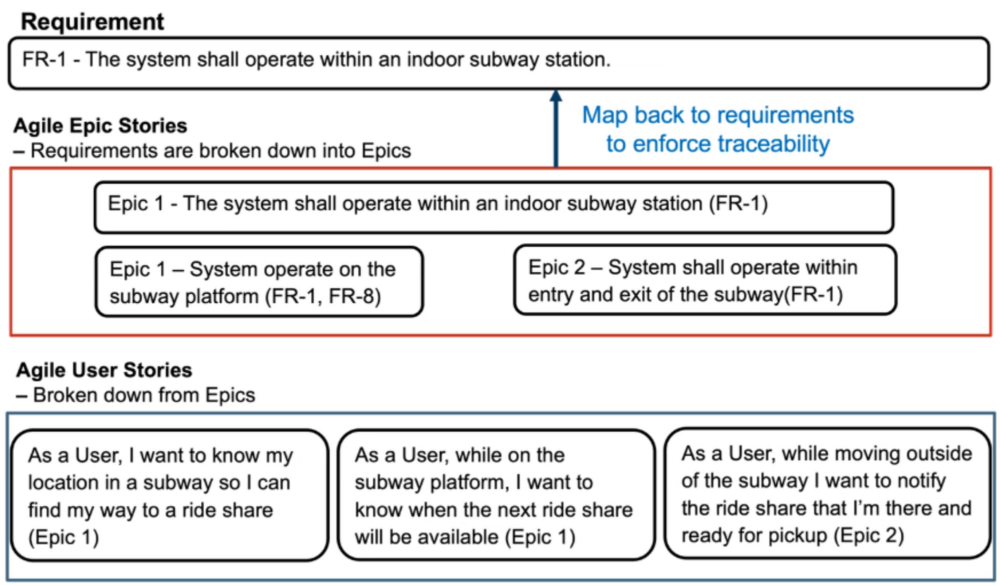
Source: ARC, 2021

Figure 9. Agile (Scrum) Process

Step 1: Software Product and Release Planning. Content from the Vee-Process deliverables will serve as inputs to the Agile Process. Using these documents as inputs, the ST-CTN Development Teams will work with ST-CTN Stakeholder Groups to perform Software Release Planning. During Software Release Planning, user needs and requirements will be broken down into Epics and Agile User Stories. Epics are large user stories, typically one that is too big to fit in a single Sprint. Epics need to be broken down into smaller user and system stories before implementation as part of a Sprint. User Stories include a statement (or a group of statements) that expresses the desired end user functionality. User Stories are simple, short, and easy to implement. User stories will be expressed in a simple sentence, structured as follows:

"As a [persona], I [want to], [so that]."

An example of mapping requirements to epics and user stories is provided in **Figure 10**.



Source: USDOT, 2021

Figure 10. Mapping Requirements to Epics and User Stories

User Stories will be captured in the Product Backlog which will be kept in the ST-CTN Agile Development Process Tool (e.g., Jira). The Backlog will be organized as a roadmap, an ordered list of the work to be done in order to create, maintain and sustain the ST-CTN products. It will be managed by the Scrum Master with direction from the Product Owner and input from stakeholders and will be visible to all team members depicting a real-time picture of the work that the Development Team plans to accomplish during the Sprint.

Once the SDD is created, the team will review the user and system stories and acceptance criteria in the SDD to develop a Product Roadmap and Release Schedule, which is essentially a phased deployment schedule that can be shared with the project stakeholders. The Product Roadmap will include a high-level plan that shows when in the future new products are expected to be developed or introduced by the team. The Product Roadmap will define and present major releases of the ST-CTN system. The Product Roadmap will capture the schedule for releases including the MVP. The Product Roadmap will also identify times when all software products and hardware will be integrated together. The Product Roadmap will guide the ST-CTN project team in Sprint Planning to ensure that key milestones are met. The Roadmap will be reviewed and updated periodically (i.e., quarterly) with the Product Owner, Development Teams, and Stakeholder Groups.

The *Definition of Ready* and *Definition of Done* will also be established at this point. The *Definition of Ready* includes the conditions that need to be satisfied by the Product Backlog item before it is considered ready to pull into a Sprint during Sprint Planning. Ready means that the story is ready to be used in a Sprint and will include the following:

- User Story is defined and complete
- User Acceptance Criteria are defined
- User Story dependencies are defined
- User Story is sized correctly by the Development Team
- The person who will implement the User Story is identified; the team members who will conduct the technical and hygiene reviews are identified.
- The team has a good idea what it will mean to Demo the story

The ***Definition of Ready*** is encapsulated in the Development Design Template. The Sprint Team will review and ensure that the template is completed and meets all the aspects not only of the story, but the components that enable the function to operate correctly and as expected. The development design template is described in **Table 5**.

A clear ***Definition of Done*** will also be established that includes an agreed upon set of items that must be completed before a project or user story can be considered complete. The Definition of Done will consist of a checklist of items such as:

- Acceptance criteria is demonstrated to the Product Owner on the production environment
- Code is peer reviewed and is checked in
- Code is documented
- Code is merged back into the main branch
- Version number is updated, if needed
- Unit testing has been completed
- Integration testing (as needed) has been completed
- Development Design Template is reviewed to ensure completion of all identified activities.

Step 2: Sprint Planning. Sprint Planning consists of processes related to planning and estimating tasks for an upcoming Sprint. Prior to beginning a Sprint, user stories and their related acceptance criteria will be created and incorporated into the Prioritized Product Roadmap. The Product Owner, with support from the Scrum Master and the rest of the development team will be responsible for prioritization of the Roadmap. The Product Owner will have the final word on the prioritization. Acceptance criteria include the product characteristics, specified by the Product Owner, that need to be satisfied before they are accepted by the user, customer, or other authorized entities. These are used to measure and compare the characteristics of the final product with specified characteristics.

Once all stories are ready, a Sprint Planning meeting will be held with the team to create a Sprint Backlog containing all tasks to be completed in the Sprint. The team will consist of the Product

Owner, Scrum Master, and Development Team, and other team members relevant to the Sprint. Other relevant team members may include USDOT, IOOs, end users, etc. as is necessary based on the focus of the development. The purpose of the Sprint Planning is to reach agreement with the team members that they have time to complete the tasks assigned to them for the sprint. The Scrum Master and ST-CTN Development Teams will estimate the effort required to develop the functionality described in each user story, and the ST-CTN Development Team will commit to delivering the tasks during the Sprint period.

Step 3: Implementation. Implementation includes the execution of the tasks and activities in a Sprint to iteratively create the ST-CTN product. In this step, the ST-CTN Development Teams will work on the tasks in the Sprint Backlog to create deliverables, features, or products. The ST-CTN Agile Development Process Tool, such as Jira, will be used to track the work and activities being carried out. Everyday a highly focused, time-boxed meeting (or Daily Standup) will be conducted. These meetings will serve as the forum for the team to update each other on their progress and any impediments they may be facing.

Key activities included in this step include:

- **Development.** Development includes confirmation of the technical approach, building features, and/or conducting unit and/or integration tests. The Development Team will work to deliver an MVP in accordance with the Product Backlog. The team will then incrementally develop additional features to deliver the full product.
- **Code Review.** Review includes conducting two types of code review (functional and technical) and code hygiene (style and documentation).
- **Documentation.** Documentation includes developing appropriate documentation as described in Section 3.1.8 and uploading materials within the appropriate Agile development tools. The SE documents, their development and review processes are detailed in the sections below.
- **Test and Deploy.** This activity includes merging features back into the main branch, conducting internal testing, identifying and fixing bugs (if required), and conducting acceptance testing. Acceptance tests will be used to check the requested and implemented feature and determine whether these deliverables, features, or products meet the requirements. Rather than taking a big-bang approach to testing, testing will occur incrementally as part of the Agile Process. All user stories will be mapped to user needs and requirements – and will include acceptance criteria and test cases.

Step 4: Review and Retrospective. This step includes reviewing the deliverables, features, and products and the work that has been done and determining ways to improve the practices and methods used to do project work. The ST-CTN Development Team will demonstrate the Sprint deliverables to the Product Owner and stakeholders in a Sprint Review meeting. Where appropriate, proxy users – or representatives from stakeholder groups – may be included in Sprint reviews to provide feedback. Discussions, actions, action items, and agreements will be documented in meeting notes and stored on the project document site. In addition, meetings may be recorded so that individuals that were unable to attend the meeting can review the recording. The purpose of Sprint Review meetings is to secure approval and acceptance from the Product Owner and/or stakeholder group for the deliverables created in the Sprint. The Scrum Master and Development Team will also conduct a Sprint Retrospective to discuss the lessons learned throughout the Sprint. This information is documented as lessons learned which can be applied to future Sprints. A feature may be sent back for refinement if the reviewers agree on an update, change or fixing a perceived flaw in the implementation.

Sprint metrics will be captured throughout the Agile Process. Scrum metrics can help teams establish benchmarks and guide the direction of the work. Tracking Scrum metrics also helps bring visibility to various dimensions of a team's effectiveness. Metrics will be captured include, at a minimum, Velocity Charts and Sprint Burndown Charts. The metrics will be presented, reviewed, and discussed during Sprint Retrospective meetings. All metrics will also be kept on the document sharing site.

3.2.1 Agile Development Tools

Several development and configuration management tools will be incorporated into the Phase 2 and Phase 3 processes to manage design, development, test and accept activities during the project. The development tools will include the following:

- **Agile Development Process tool:** An Agile planning and project management tool provides a forum to plan, track, and report on the Agile development. The tool will manage Product and Sprint Backlog, trace the user stories to requirements and the acceptance criteria to test procedures/scenarios/cases. The tool will be used to track risks, impacts, and defects. The tool will also be used to assign work to designers, developers, reviewers, and testers, and to facilitate communications among developers. The tool will also collect and generate Sprint Metrics, issues, and bug reporting. In addition, the tool will be used to collect user feedback during the Sprint Planning meetings and during the Retrospective and Demonstration meetings. Among the tools under consideration include **Jira** (Atlassian) and **GitLab**.
- **Agile road mapping / scheduling tool:** The ST-CTN project team will employ a road mapping tool that allows developers to map and visualize user stories for consumption by stakeholders. The decomposition enables developers to understand the dependencies and collaboration methods to manage the workflow. Among the visualization tools under consideration are **GitLab** and **Monday.com**. The SDS will be incorporated into the Master Schedule which uses **Microsoft Project**.
- **Document management tools**: The ST-CTN project team will utilize document management tools to manage and ensure configuration control of the system engineering documents. **Microsoft tools** will be used to generate the documents stored in the **Teams / SharePoint** to ensure configuration control. Details of the document management naming conventions and version control will be included in the Phase 2 Program Management Plan. The source code, tests, and development environments will be managed through the use of subsystem developer configuration tools.

Re-entering the Vee-Process for Verification and Validation

Once all products have been delivered and accepted, the Agile (Scrum) Process ends and the process goes back to the Vee-Process for verification and validation testing. System Verification testing consists of executing the complete set of system integration tests, checking that all user stories are mapped to user needs and requirements, including acceptance criteria, test cases, and test results. This information will be documented throughout the Agile Process in an Agile

Development Process Tool, such as Jira and will be exported to a spreadsheet to show traceability. The matrix will link to applicable testing documentation used during the Agile Process.

The information created throughout the Agile Process will be captured in a comprehensive STP. The STP will document the specific test plans, test procedures, and test cases that were used during the Agile Process. The STP will also include a Requirements Traceability Matrix (RTM) that shows the traceability between every user need, system requirement, epic, user story, test verification method (Inspection/Demonstration/Analysis/Test) and the specific test case(s), as shown in **Table 3**.

Table 3. Example Traceability Matrix

User Need	Requirements	Epic	User Story	Test/Acceptance Criteria	Test Results	Status
User Need 1	Requirement 1.1	Epic A	User Story 1 User Story 2	US1 – Acceptance Criteria US2 – Acceptance Criteria	US1 – Test Results US2 – Test Results	Accepted Accepted
User Need 1	Requirement 1.2	Epic A	User Story 3 User Story 4 User Story 5	US3 – Acceptance Criteria US4 – Acceptance Criteria US5 – Acceptance Criteria	US3 – Test Results US4 – Test Results US5 – Test Results	Accepted Accepted Accepted
User Need 2	Requirement 2.1	Epic B	User Story 6 User Story 7	US6 – Acceptance Criteria US7 – Acceptance Criteria	US6 – Test Results US7 – Test Results	Accepted Accepted
User Need 2	Requirement 2.2	Epic B	User Story 8	US8 – Acceptance Criteria	US8 – Test Results	Accepted
…	…	…	…	…	…	…

As one of the final demonstrations that the ST-CTN system is ready to enter operations, an ORT will be conducted. The ORT will be documented in the Operational Readiness Test Plan (ORTP). The ORTP will detail plans (specific requirements listed later in this section) for a series of coordinated tests and demonstrations used to ensure the operational readiness of the system. The objectives of these activities are to demonstrate the deployed system operates as designed in a safe and secure manner before operational deployment begins. Operational readiness conceptually applies to the system itself as well as the implemented institutional and financial framework that supports, finances and governs the deployed system.

Once the system is fully accepted, operations and maintenance will begin based on a CMOP developed by the ST-CTN project team. The document will summarize key operational methods and procedures that ensure safe and efficient operations in Phase 3, incorporating as required, elements from the SMP. It will also include a high-level maintenance approach, as well as a high-level plan for inventory and configuration management.

3.2.2 User Needs Processes

The needs provide a foundation for all subsequent systems engineering processes and are of critical importance to the success of the ST-CTN project. A user need is an expression of a required capability of the system, expressed in generic (i.e., non-solution oriented) terms. System needs, system requirements, and design elements are traced to identified user needs, so developing a thorough list of user needs at the onset of the project allows all subsequent processes to flow smoothly.

The ST-CTN project team developed a UNIRP document to define the processes that were used to generate, coordinate, approve, and support the configuration control of user needs and system requirements. The processes defined in the UNIRP document provided the foundation for how user needs and requirements was developed.

The purpose of the Needs Summary is to document, organize, and understand user needs that were collected as part of the stakeholder engagement process for the ST-CTN project. The steps used to develop User Needs are summarized below with additional details included in the UNIRP.

Step 1. Preliminary User Needs from the ST-CTN Proposal

When developing the ST-CTN proposal, the project team used existing master plans and outreach reports, combined with a thorough knowledge of the project area and needs of underserved travelers to gain a preliminary understanding of user needs. These high-level needs led to the development of the preliminary concept of the ST-CTN system and have been used to provide guidance to the project team throughout the systems engineering process.

Step 2. Stakeholder Engagement

To transform these high-level user needs into well-written user needs, the ST-CTN team conducted stakeholder outreach to underserved communities, owner/operators, and system developers to:

- Gain additional insight into what specifically generated the high-level needs
- Understand which needs are of major importance to the stakeholders
- Ensure that the needs are not currently solved by existing system features
- Understand why the capabilities to address these needs are required in the system

The ST-CTN project team held a series of three work sessions to identify a comprehensive list of stakeholders who serve or represent end users, own or operate infrastructure, or are system developers for subsystems within the proposed system deployment area. These stakeholders were then categorized by their area of expertise. The ST-CTN project team conducted user and system interoperability needs interviews with the following stakeholder groups:

- **Underserved Communities.** Advocacy groups representing underserved communities including users with cognitive or physical disabilities, aging adults, and persons with LEP.
- **Infrastructure Owner/Operators (IOOs).** Public or private agencies that will set policy regarding the deployment and maintenance of the deployed infrastructure and services.
- **System Developers.** Organizations and companies that will develop the software responsible for collecting and distributing the information needed in the final system deployment.

Step 3. Documentation of User Needs

Inputs from the interviews served as the basis for the ST-CTN project team to develop user needs that were documented in the Needs Summary. The ST-CTN Needs Summary, identifies and describes needs for the ST-CTN system. The purpose of the Needs Summary was to document, organize, and understand user needs that were collected as part of the stakeholder engagement process for the ST-CTN project. The Needs Summary document focused on the end user needs of the system and will be used to guide the development of the institutional, data security, and system interoperability needs which is presented in the ConOps.

User needs were collected through interviews with representatives for the underserved communities. The ST-CTN team conducted focused interviews with stakeholders to gain an understanding of existing challenges and user needs. The ST-CTN chose to conduct interviews with small, targeted stakeholder groups to maximize engagement and to ensure that stakeholders are given an opportunity to convey their challenges and needs to the project team.

Step 4: Use Case Decomposition

When incorporated into the ConOps, high level use case descriptions were developed to align with the user needs. During the interviews, stakeholders were asked to describe travel scenarios tied to the Traveler Complete Trip segments where they encounter barriers and tools and services that would aid a seamless journey. These operational scenarios will be used to generate use cases to illustrate user needs and highlight information flows between subsystems and components in the ConOps. The needs that emerge from stakeholder interviews and meetings will be documented in the Needs Summary. The Needs Summary document will include well-articulated needs with supporting rationale. This summary will be reviewed with the Stakeholder Panel (a selection of key personnel from each stakeholder group) and incorporated into the ConOps.

3.2.3 Concept of Operations Processes

The final needs were developed within the ConOps – and have been put under configuration management. These needs are being used to guide system requirements and ultimately used to validate the proposed system design. The ConOps used the template provided by the U.S. DOT that is based on *IEEE 1362-1998 – Guide for Information Technology—System Definition—Concept of Operations (ConOps) Document*. The ConOps is a foundation document that frames the overall system and sets the technical course for the project. Its purpose is to clearly convey a high-level view of the system to be developed in a way that every stakeholder can understand. The ConOps answers who, what, where, when, why, and how questions about the project from the viewpoint of each stakeholder.

Similar to User Needs, the process for developing ConOps was documented in the UNIRP document. The ConOps development processes consisted of:

- Integrating the User Needs from the User Needs Processes (**Section 3.2.2**)
- Drafting the ConOps that was guided by IEEE 1362-1998.
- Soliciting input from stakeholders (end users, IOOs and system developers, as well as other interested parties) by facilitating a ConOps Walkthrough.
- Updating the ConOps to incorporate stakeholder input from walkthrough.

Additional details pertaining to the process used to develop the ConOps are included in the UNIRP and ConOps.

3.2.4 Requirements Processes

One of the most important attributes of a successful project is a clear statement of requirements that meet the stakeholders' needs. System requirements are the technical statements that define the capabilities that the system must provide or constraints that the system must operate within. The ST-CTN project team understands that much like user needs, identifying and developing solid system requirements is critical to the success of the overall program. System requirements must also be traceable to one or more user needs. Documenting and tracking this traceability is an important element to ensuring the system ultimately meets all of the user needs identified by the stakeholders.

Similar to User Needs, the process for developing system requirements was documented in the UNIRP document. The steps used to develop requirements are summarized below with additional details included in the UNIRP and SyRS documents.

Step 1 Requirement Decomposition

Starting from the use cases, the user, IOO and system needs were decomposed into unambiguous and concise requirements. Attention was paid to identify requirements relative to several key categories required to specify the system. The categories covered the following groups: Functional, Infrastructure, Data, Operational Performance, System Operations (reliability, scalability), Performance Measurement, System Security and Privacy, Information Management, System Operations and Maintenance, System Lifecycle Sustainment (technical and non-technical requirements) and System Interfaces.

The ST-CTN project team held technical work sessions with project stakeholders in order to develop system requirements for system integration and specific subsystems. In particular, technical work sessions included IOOs and system developers. User, IOO and system needs from the ConOps document were used to drive the conversations during these meetings. The ST-CTN project team expand each need into a set of high-level requirements which covered the categories listed above.

Step 2 System Requirements Document Development

The ST-CTN project team developed the system requirements in accordance with IEEE Standard 1233-1998 and the SEBOK. The ST-CTN project team held technical work sessions with project stakeholders in order to develop system requirements for system integration and specific subsystems. Technical work sessions included IOOs and system developer project stakeholders. User needs and system needs from the ConOps document were used to drive the conversations during these meetings.

The ST-CTN project team expanded the high-level requirements to more detailed requirements which were assigned to a particular category. These categories were used to determine completeness, verifiability, and feasibility. Once developed, internal reviews of each category and related categories were conducted by system developers and IOO stakeholders prior to being reviewed by USDOT and the SyRS Review Panel.

Step 3 System Requirements Traceability

Each requirement was indexed, thoroughly documented, and traced to the user needs throughout the systems engineering process. A RTM was developed and stored in an Excel workbook which documents needs, requirements, and their relationship. This workbook will be used as the master document to store, update, and trace needs to requirements, requirements to design elements (e.g., user / system stories), datasets, code, acceptance criteria and test cases that validate and verify that the system works and meets user needs.

The RTM includes the following columns described in **Table 4**.

Table 4. Requirements Traceability Matrix Set Up

RTM Field Name	Description
Requirement ID	A unique identifier for each requirement
Subsystem	The subsystem name for which the requirement is associated. Only include a subsystem name (e.g., ATL RIDES) if this is a unique requirement for the integrated subsystem application.
Requirement Category	The type of requirement category.
Need ID	The unique identifier that corresponds to the needs defined in the ConOps.
Need Description	The user need description that corresponds to the need defined in the ConOps.
Requirement Title	A title for the requirement.
Requirement Description	A description for the requirement.

RTM Field Name	Description
Related Need, Requirement, DMP ID	Any related needs, requirements, dataset identifiers, or other enumerated system engineering item or artifact
Verification Test	Describes how the requirement should be verified. The verification categories include: observation, compliance test, or test.
Date Inserted	The original date wherein the requirement was inserted into the spreadsheet.
Inserted By	Initials of the person who inserted the original requirement.
Date Updated	A date when the requirement was last updated.

The lineage information in the table will enable version control of the requirement. Additional information will be included to augment configuration management of the requirements.

During subsequent processes, the RTM will be augmented to include information that drives the Agile Product Roadmap including if the feature / function exists (e.g., open trip planner feature), the gap, the user story(ies), acceptance criteria, and test case(s) that relate to the requirement.

Step 4 System Requirements Walkthrough

The SyRS walkthrough has two major purposes:

- Review and confirm that each requirement traces to no more than one user need and to ensure that the requirements have been developed to address all user needs.
- Verify and validate that the requirements are well-formed, complete, consistent and prioritized.

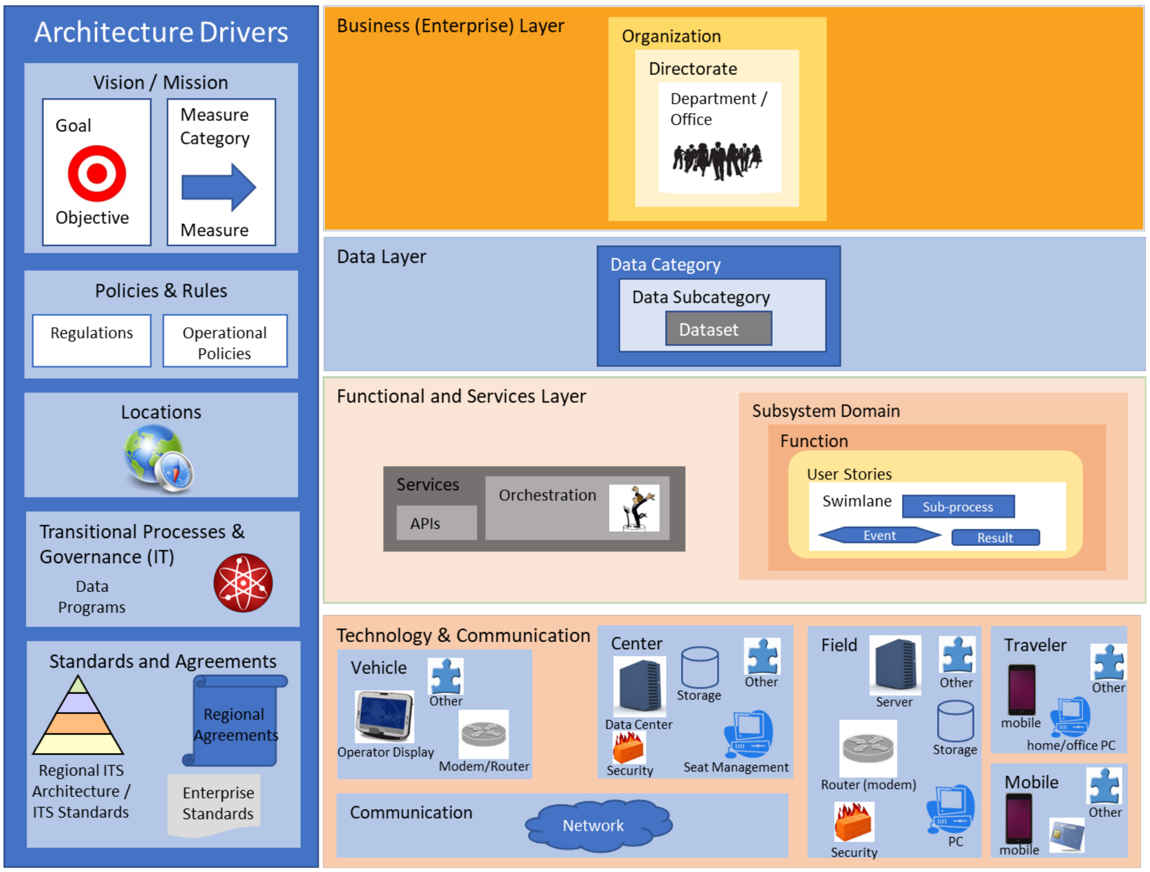
The SyRS walkthrough was organized to map requirements to user needs. The facilitators led the review and discussion to verify the traceability of user needs to requirements; well-formedness for each requirement, and completeness of requirements to meet augmented component functionality, interface communications, data integration, and other factors that ensure proper operations.

Step 5 Finalization of System Requirements

Following the walkthrough, the SyRS and RTM were updated to reflect stakeholder input into the document.

3.2.5 Architecture Processes

The ST-CTN system is composed of various interconnected layers that form a coherent enterprise architecture. The architecture provides a blueprint for understanding the entities, components and relationships among the people, processes, and rules. A metamodel or actors that compose the model are illustrated in **Figure 11**. The layers comprise the four architectures identified in the Phase 2 and 3 Program Tasks. To be complete, four layers and a set of architecture drivers are identified in **Figure 11** that comprise the ST-CTN architectural processes. The processes related to each layer are discussed below.



Source: adapted from TCRP Report 84, P. Okunieff, 2011

Figure 11. Architecture Metamodel

Enterprise Architecture (Business Layers): The Enterprise Architecture provides information on the **Business Organizations** responsible for elements of the system, as shown in **Figure 11**. The organizations establish the roles and responsibilities for operating, maintaining and sustaining the system. Specifically, the enterprise architecture will trace to system lifecycle sustainability requirements, as well as the roles and responsibilities, constraints, and impacts described in the ConOps. The enterprise architecture will address the following questions (extracted from the ARC-IT Version 9.0):

- Who is responsible for providing transportation-related user services?

- Who is responsible for installation, operations and maintenance of ITS services, applications and devices?
- What relationships need to exist between transportation operators to facilitate services between and within jurisdictions?
- What relationships need to exist between the providers of services and the consumers of services?

Data Architecture (Data Layer): The data layer is defined in the DMP. The Phase 1 DMP included dataset types that correspond to the data categories detailed in the data layer. Data schema, specifications, definitions and standards are associated with the datasets included in the data layer. In addition, the protected nature of the datasets is also included in the DMP and will inform the related functional and physical architecture entities. The layer will link to the specific datasets in the most current DMP. The data layer will be updated when the Phase 2 DMP and Updated DMP (Phase 3) documents are reissued.

Function and Interfaces Architecture (Function and Service Layer): The current context diagram (**Figure 4** and described fully in the ConOps) and Phase 1 DMP identify subsystem modules and information flows (including APIs, specific datasets, their type, and related standards). The Function and Interface Architecture will be refined to reflect details and updates to the functions and services in the ST-CTN system. The architecture functional entities will identify the high-level user stories that are derived from the SyRS for each subsystem. The Design Processes will link the detailed user stories in the Product Backlog to the functions in this layer.

Physical Architecture (Technology and Communication Layer): The Physical Architecture will use the ARC-IT Version 9 physical entities to describe the assets used to implement the ST-CTN system. The physical architecture answers these questions:

- What physical entities are involved in the delivery of a given service?
- What interfaces are required between different physical elements?
- What functionality is allocated to physical entities?
- What are the security considerations for information exchanged between physical elements?
- What are the security considerations for physical devices?

Summary diagrams of the Service Packages that correspond to the functions will be included in the physical architecture. Several ARC-IT Version 9 service packages include gaps in related Multimodal and Accessible Travel service packages. The ST-CTN project team will work with the USDOT to propose additional classes, actors, and information flows to complete the services provided by the ST-CTN project.

Communications Architecture: The Communications Architecture is a subset of the Technology and Communication Layer.

The SAD development processes include:

Step 1: Draft system architecture layers (SAD)

- Using existing legacy subsystem architectures, updated Phase 1 System Engineering documents and the Phase 1 ICDTP, the EMT and other technical team members will hold technical meetings to model the relationships associated with each architecture layer as well as between layers.
- Develop the four architecture layers and document them in the Draft SAD.
- Review draft SAD by the QA engineer for quality and technical issues.

Step 2: Review system architecture layers

- Provide draft SAD to the USDOT for review.
- Receive Comment Matrix from USDOT with document comments.

Step 3: Organize a walkthrough of the SAD

- Identify and invite participants.
- Develop and distribute a walkthrough workbook.

Step 4: Facilitate a walkthrough

- Facilitate a walkthrough of the Draft SAD guided by IEEE Standard 1028-2008.
- Solicit and record comments and recommendations.
- Distribute walkthrough comments to participants.

Step 5: Update SAD

- Incorporate comments into final Phase 2 SAD.
- Submit final SAD and document comment matrix response to USDOT.

3.2.6 Design Processes

Based on the SyRS and the SAD, the system design will be created describing the full scope of the system. The system design will consist of an initial Product Backlog and schedule for the ATL RIDES and STM (including performance measurement dashboard components) subsystems and the systems that integrate with it, including the CV subsystem elements and beacons. The system design will be presented as a high-level design document (i.e., a preliminary design,) which includes a preliminary set of user and system stories, database, process workflow, user interface (UI), and dependencies (including interfaces).

The preliminary SDD will be created to guide the Agile Development process. Requirements will be allocated to the user stories, system components, and interfaces will be specified. The project team will develop the preliminary SDD at the onset of Phase 2. IEEE 1016-2009: Recommended Practice for Software Design Descriptions will be used as a guideline for format and content to develop the SDD. A walkthrough of the preliminary SDD will be held with stakeholders and the USDOT to refine the user stories, priorities, process flow and user interfaces, and to identify system dependencies.

During the Agile Process, design and design reviews will be conducted as part of Sprints. All design artifacts will be documented and stored on MS Teams and mapped back to user stories, requirements, and user needs in the RTM. This will ensure traceability between user needs, system requirements, and system design.

Rather than taking a traditional systems engineering approach and developing a detailed design document at the on-set of the design phase, the Draft and Final SDD document will be iteratively developed. The Agile development process will be leveraged to facilitate the iterative design, review, deploy, test, and documentation of system components. The artifacts created during the Agile Process will be packaged up at the end of the development process to create an As-Built Draft and Final SDD versions. The ST-CTN project team will submit the preliminary SDD to the USDOT to review the process and expected detail of the document. The project team will submit the as-built Draft SDD following aggregation of the design artifacts iteratively generated through the Agile Process. The design will be completed iteratively throughout the process. The iterative design artifacts will be developed and reviewed throughout the Agile Process and will be available to the USDOT upon request. At the end of the process, the project team will submit the as-built Draft SDD following aggregation of the design artifacts iteratively generated through the Agile Process. USDOT will review the Draft SDD and provide comments to the ST-CTN project team on format and presentation. The ST-CTN project team will address USDOT's comments prior to delivering the Final SDD.

The following processes will be implemented to develop, review, and finalize the preliminary SDD:

Step 1: Develop preliminary SDD

- Confirm requirements
- Determine gaps in existing features
- Develop user and system stories
- Plan acceptance criteria to be used for unit and integration tests

Step 2: Review preliminary SDD with Stakeholders

- Set up SDD walkthrough and distribute walkthrough workbook to stakeholders including the USDOT AOR.
- Hold walkthrough and solicit comments on prioritized list of features, refinement of user stories and acceptance criteria
- Validate user stories are consistent, complete and map to user needs and requirements

Step 3: Update SDD / Product Roadmap

- Update Product Backlog and Roadmap
- Update preliminary SDD incorporating stakeholder input and modifying process to align with documentation requirements of the Draft and Final SDD

During the Design Sprints, the approved Sprint Templates (see Table 5) will contribute to the Final SDD. The completed templates will be reviewed during the Sprint Review meetings and deemed ready for development and implementation by the Product Owner. They will be reviewed and updated throughout the Agile processes if impacts or issues are identified. The Draft and Final (as-built) SDD versions will be a compilation of all the templates into a final design document.

3.2.7 Development and Implementation Processes

The development and implementation processes are initiated during Sprint Planning meetings. These processes form the core of the Agile Process. Input from the Prioritized Product Roadmap, Product Backlog, Sprint Review, as well as stakeholder groups and ST-CTN project team will drive the selection of Sprint activities. The Scrum Master will work with the development team to select user/system stories that are ready for development. As noted in the overall process descriptions, a Sprint is ready when the Development Design Template is completed and reviewed by the Sprint Team.

The Development Design Template is described in **Table 5**. The template serves as a checklist to use when starting a user story / software development task. The template ensures that a minimum level of quality can be achieved without having to get the sign-off on changes from everyone within the team. The template also implements the SDLC processes that are implemented during software lifecycles.

Table 5. Development Design Template

Development Design Template Categories
Task Assignees. Primary developer: First reviewer: Final reviewer:
Task Documentation. A new document should be created for each task. This document will be used to include notes about changes to various aspects of design and any decisions that were made along the way or that describe the current status of this task.
User Stories / Functionality / Spec. Key questions to answer, before starting, include: • Has there been at least one sentence written out describing what this task should result in accomplishing? • Does that one sentence sufficiently explain the complete functionality to be implemented? If not, add more detailed stories that describe smaller pieces of functionality until everything is sufficiently explained.

Development Design Template Categories
System Design. Does the proposed change create any new systems or significantly modify any existing systems or how these systems communicate with each other? If so, create a sketch of all of the systems involved and how they interact. Get sign-off on these changes during a weekly technical call with the whole team so that the full team can be aware of these changes.
Process Workflow. Key questions to answer, before starting, include: • Does the proposed change involve creating or modifying a fairly complex sequence of events or data processing? At the very least, have a meeting before writing the code with the pull request review team members to make sure everyone is on-board with the changes. • Database Design. Does the proposed change create or modify any parts of a data model which gets persisted to a database? If so, write up the proposed changes and get sign-off on these changes during a weekly technical call with the whole team.
User Interface. Key questions to answer, before starting, include: • Does the proposed change significantly modify the UI? If so, create mockups of the proposed changes and receive buy-in about the changes from the pull request review team members (i.e., Development Team and Product Owner). If the changes are very significant, also receive buy-in during a weekly technical call with the whole team.
Tests. Key questions to answer, before starting, include: • Does the proposed change or sub-parts of the change involve a lot of different input scenarios? It is highly recommended that tests are made for as many of these scenarios as possible. Ideally, test cases should align with the user stories so that automated code can verify that the resulting software properly implements the user stories. • All repositories should have GitHub Actions setup that run at a minimum on pull requests. Repositories with JavaScript must perform linting (checking code for errors) as a step within the GitHub actions script and should strive to have TypeScript support. Repositories with Maven should run the \`mvnpackage\` command in one of the steps to see if the project can be built and tests run.

Development Design Template Categories
Configuration. Key questions to answer, before starting, include: • Does the proposed change require any changes to configuration? If so, list out what is proposed to be changed and receive buy-in about the changes from the pull request review team members.

Once the process is implemented, the template will include locations and links to documentation for each category. The Sprint Planning, Daily Sprint Check-in, and Sprint Review meetings provide the major processes associated with Development and Implementation.

The development processes consist of building the feature, unit testing, and feature branch integration testing. The feature is integrated into a branch code set to perform the unit and integration tests. These are conducted prior to and following peer review processes.

The template identifies System Design elements, calling out the need to document new interfaces, modules, and system workflows. All updates to the existing code base and new development will be documented and stored in the GitHUB OSS site as described in Appendix C. The ST-CTN team is committed to open-source development for new interfaces and enhancements. As such, all new interface and enhancements will be open-source and made available on GitHub.

(Note: description of the test plan – test procedures, test cases and test data content are included in the **Section 3.2.8** and **Section 3.2.9**)

The SDLC includes two code review processes: technical review and hygiene review.

The **technical review** validates the code integrity, completeness, and verifies that it works correctly according to design specifications. The technical review examines the integration with the database and UI impacts. In addition, it will assess impacts and performance related to interfaces with external and subsystem integration.

The **hygiene review** ensures that the code style is clean and well documented, that it is clear and consistent with other code artifacts of the same subset.

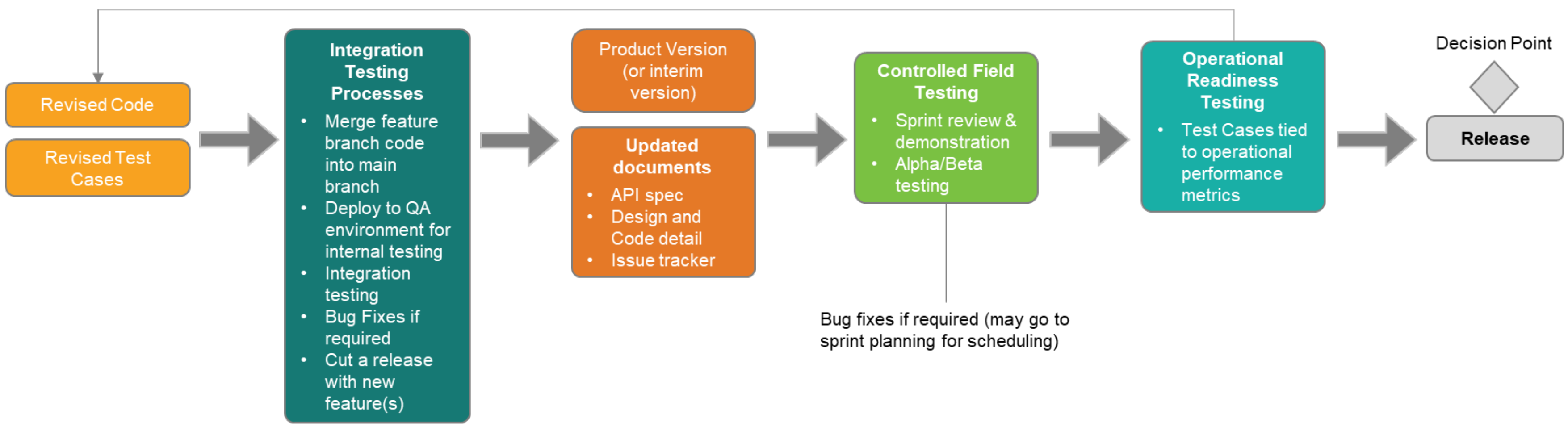
Following the code reviews, the primary developer will update the code and rerun the unit and integration tests.

3.2.8 Integration and Testing Processes

The Integration and Testing Processes begin after development, review, module (unit) testing, and feature integration testing is conducted during the Development and Implementation processes. Integration and testing processes implement main branch testing processes. Testing conducted during the main branch testing processes will provide the documentation of the STP and Results.

3.2.8.1 Integration and Testing Process Flow

During the Development and Implementation Processes, as noted in **Figure 9**, the unit and integration tests procedures are built. These test procedures include development of and sequence test cases, testing criteria, testing methods, and test data to be conducted during the tests. Integration testing for the main branch code will include not only the new integration test cases, but a subset of all the test cases generated for the existing code base to ensure that the additional features and functions did not impact other parts of the code. These tests will be documented, reviewed, and signed-off by the systems engineering team and the Product Owner. In addition, all graphical user interfaces for web and mobile sites will be tested for functionality (through automated scripts) and accessibility by useability experts.



Source: ARC, 2021

Figure 12. Testing Processes

3.2.8.2 Test Environment

A test environment will be set up to test main branch merging and testing. The environment will include test data generators and simulators for external system and interface exchange. As multiple subsystems are deployed, test environments set up to test stand-alone features will be integrated into a more complete ST-CTN test environment.

3.2.8.3 Test Plan Repository

All test plans including test procedures related test cases, test data, and links to their code base versions will be stored in a GitHub site (as described in **Appendix C**). The repository contains tests that address all the features, functions, and performance of the subsystems for the legacy and new components developed for this system. The collection will be used for regression testing during the merge process to verify that the system continues to work safely, correctly and without defect.

3.2.8.4 Test Planning

Test planning begins as soon as a feature is included in the Sprint Backlog for development. Test planning for each software feature begins with specifying the inputs, testing cases, and expected results using scenarios for how the feature handles a variety of operations or conditions. The test plan will include information on the feature to be tested, test documentation, test verification methods, pass/fail criteria, testing roles and responsibilities (including review, test, certify), test deliverables, and schedule.

Test Data

Test data will be developed to test how the feature reacts to varying input conditions. The conditions will include the following testing methods:

- Positive (or valid) testing – input values should be processed successfully.
- Negative (or invalid) testing – input values should cause the system to execute appropriate error processing, such as an error notification message to a user.
- Boundary values testing - including those just above, just below, and precisely on each limit.

Test Procedures and Scripts

A test procedure is defined as a set of test cases comprised of step-by-step instructions used to exercise each flow and process implemented in a function. The test procedure may be documented for manual execution or written as an automated script. Test procedures are defined by specifying test cases, trigger events, and expected results with pass / fail criteria included for each test case. Automated scripts can run through multiple test data and generate exception reports which detail anomalies from expected outcomes. Test procedures and their related test cases will be reviewed by Sprint Team members prior to execution to ensure they meet requirements and acceptance criteria reviewed by the stakeholders during the Agile Planning / Review meetings.

The test procedure will run multiple times with different test data to exercise the feature over varying conditions.

Test Pass / Fail Criteria

The **Testing Criteria** specifies the conditions to be used to determine whether each test case passed or failed testing. For Main Branch **integration testing**, the pass criteria include:

- Data values are correct and within specified value ranges.
- Data structures are correct.
- Error handling is correct.
- Performance criteria are within specified time limits.

For **controlled field testing** (demonstration, alpha and beta testing), the pass criteria include:

- Number of anomalies and anomaly types do not exceed limits over a period of time.

For the **operational readiness testing** (acceptance testing), the pass criteria are tied to performance metrics of the project.

If a test failed and instigated a requirement change, the systems engineering team will track the changes throughout the system engineering documents – SyRS, SAD, SDD and related documents (e.g., DMP, SMP, PMESP), and mark up those documents with input from the technical team. Updated features will need to be developed and test cases rewritten to address the change in requirement, design, and code.

Test Procedure Review

The test procedures, data, cases, tests criteria and scenarios will be reviewed by the technical team to address the following:

- Does the proposed change or sub-parts of the change involve a lot of different input scenarios? It is highly recommended that tests are made for as many of these scenarios as possible. Ideally, test cases should align with the user stories so that automated code can verify that the resulting software properly implements the user stories.
- Do all code repositories have GitHub actions setup that run at a minimum on pull requests? Repositories with JavaScript must perform linting as a step within the GitHub actions script and should strive to have TypeScript support. Repositories with Maven should run the `mvnpackage` command in one of the steps to see if the project can be built and tests run.

3.2.8.5 Test Results

Following execution of the test procedures, the results will be reviewed by the System Engineering Team to verify that the tests are complete, validated against the requirements, and verified to run correctly. In addition, the System Engineering Team will verify that the documentation is updated and published to the test plan repository.

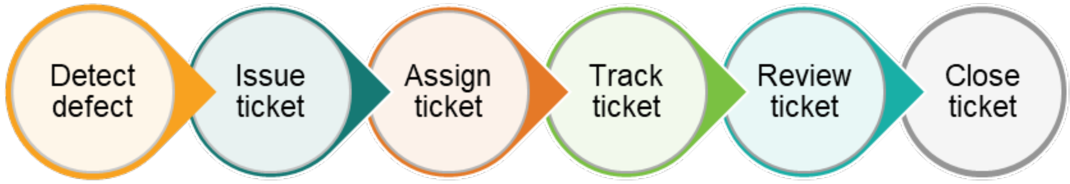
3.2.8.6 System Test Plan

Prior to release, the collection of test cases associated with the subsystem extensions will be published in a STP compliant with ISO/IEC/IEEE 29119™ Software and systems engineering - Software testing (standard series). The STP test cases and results will be used to show evidence of comprehensive testing in the ORTP, as described in **Section 3.2.9**.

3.2.8.7 Defect/Discrepancy Processes

During the testing stages including Controlled Field Testing, a Defect and Discrepancy Process will be implemented to resolve any defects or discrepancies identified during ST-CTN system use. The operations and maintenance processes will implement an incident management process to manage defects, issues, and risks.

The Scrum Master will report to the EMT on the metrics of defects, by quantity, type, criticality (impact), and state of the defect ticket. The “ticket” will be managed as a controlled list of actions that are at different stages. The Scrum Master will review the tickets through the Defect / Resolution during the Sprint Planning and Review meetings. This information will be made available to the ST-CTN project team including, the Product Owner, Development Teams, IOOs, and others as necessary.



Source: ARC, 2021

Figure 13. Defect / Resolution Process

The defect process, depicted in **Figure 13**, includes six stages from first detecting the issue through its closure.

- **Detect issue:** (trigger: call center complaint, app/website comment, automated alert). If low impact and can be handled outside of defect process, then it is resolved and will not be considered a defect. If critical, a ticket will be issued and addressed immediately by the Scrum Master (and assigned to a development team member for immediate resolution).
- **Issue ticket**: All tickets will be reviewed during the Sprint Planning meetings. If a UI/UX, then it will be added to the Product Backlog. If other, it will be added to the Sprint Backlog for scheduling at each Sprint.
- **Assign ticket:** New tickets will be reviewed and assessed during the Sprint Planning meetings (unless identified by Scrum Master as critical). Tickets will be prioritized (except UI/UX which will be prioritized during Product Backlog meetings) and assigned to technical staff to identify cause and resolve.
- **Track and review ticket:** Once assigned during a Sprint Planning meeting, each ticket (where state is open or in-progress) will be reviewed until resolved or determined that the issue is no longer applicable (and should be abandoned). A ticket may be escalated to a higher criticality if the issue cannot be resolved by revising or refactoring code.
- **Close ticket:** A ticket is closed when the issue is resolved (tested, accepted, and demonstrated).

The Scrum Master will report to the EMT on tickets and their states including expected time to resolve, impact, and other metrics.

During detection, the defect will be categorized by type and criticality. Defect types may be assigned to at least one of the following categories**:** UI/UX, operating environment, database/data (content, privacy), code/logic, (cyber)security, testing failures.

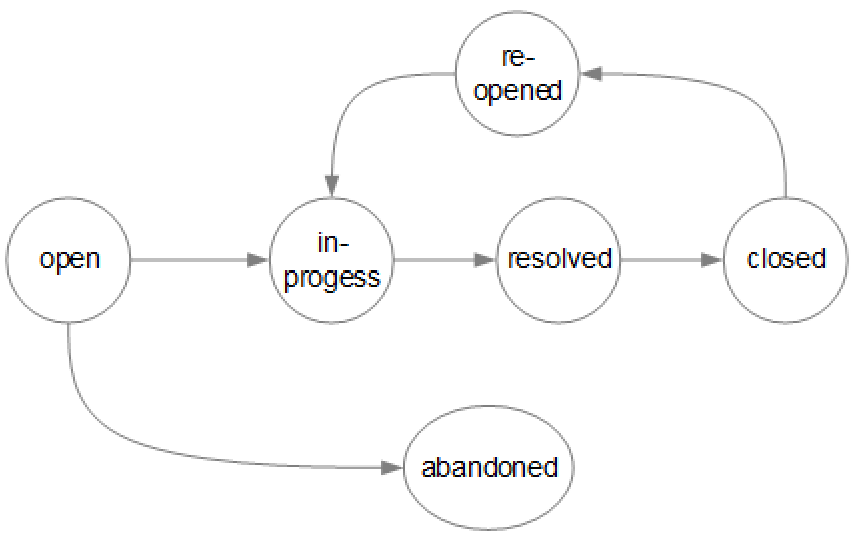
Defects are also associated with a low, medium, or high criticality level as listed below.

- Low
    - Impacts to UX: individual complaints

    - Handle through call center, frequently asked question (FAQ) or training
- Medium
    - Impacts to operations: including repeated complaints and integrated operations that does not impact safety, privacy, or security
    - Handle: triage, if cannot identify issue / origin then issue ticket with a low priority
- High
    - Impacts to participant safety or privacy (PII data), or system security.
    - Handle: triage to address immediate high impact concern then issue ticket

Once issued a ticket the defect may be assigned different states (shown in **Figure 14**), depending on its resolution. The states include:

- Open – defect is identified as an issue that must be resolved and is issued a ticket
- In-progress – ticket is assigned to a developer to resolve
- Resolved – a ticket was resolved and is waiting for final testing and approval
- Closed – a ticket received final testing and approval from the technical team
- Abandoned – a ticket was abandoned because it no longer is a problem (code was changed) or another external factor
- Re-opened – a closed ticket was reopened because the problem continues to occur



Source: ARC, 2021

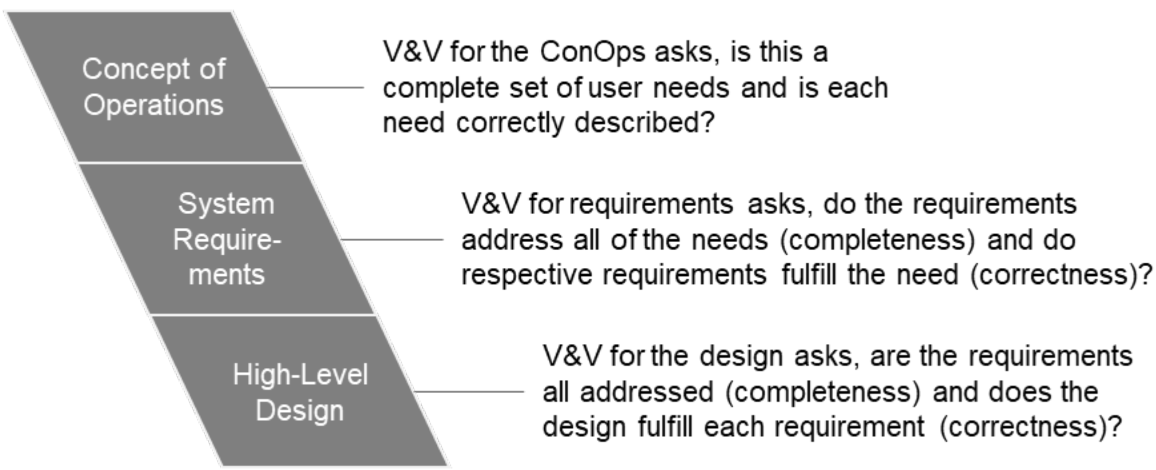
Figure 14. Ticket States

3.2.9 Verification and Validation (Acceptance) Processes

Verification and Validation is a critical activity in the systems engineering process. Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

V&V for Vee-Process Deliverables

During the System Definition phase (composed of the ConOps, SyRS and High-Level Design) V&V activities are conducted at each stage. These phase gates are used to assess if the next step in the Vee-Process is ready to begin. A project moves from one phase to the next once key deliverables are reviewed and accepted. For example, a project moves from the ConOps Phase to the System Requirements Phase once the ConOps is accepted. The Systems Requirements Phase should not begin until the ConOps is verified and validated. As an example, **Figure 15** summarizes V&V for the ConOps, SyRS, and SDD phases.



Source: ARC, 2021

Figure 15. Verification and Validation at Each Stage

For systems engineering documentation, walkthroughs play a critical role in the V&V process. These focused meetings are used to bring stakeholders to the table with the developers of the document to go through the document's key elements. This step is particularly critical for the ConOps and SyRS phases as these documents set the context for the rest of the project. Walkthroughs were conducted for the ConOps and SyRS as part of Phase 1 of the project – and will be conducted for the SAD and SDD as part of Phase 2. The ST-CTN project team will host and conduct walkthroughs in accordance with IEEE Standard 1028-2008. These walkthroughs will be used to demonstrate the completeness and technical soundness of the deliverables. Walkthrough Workbooks will be used to structure and expedite the Walkthrough process.

Agile Demonstrations

Sprint Reviews will include demonstrations. During these meetings, developers will share the work they accomplished during the Sprint. Developers demonstrates fixed bugs, new features, and infrastructure necessary for future work. The Product Owner and stakeholder groups may be invited to attend applicable Sprint Review meetings to participate in the demos and provide feedback to the development teams. In many cases, proxy end users or system users – or people representing a user group – may be asked to participate in the demonstration presentation. Users will provide feedback on the features and products presented. These comments will be captured by the development team and considered as inputs to new user stories or to update or augment current Sprint features. New user stories will be prioritized accordingly in the Product Backlog. The purpose of the end user and system user participation is to solicit incremental V&V throughout the development process. When the system is ready for release, these inputs will provide an equivalent design walkthrough, and the collection of user stories, operational flow, database, UI, and interface design elements will be published in a detailed SDD.

V&V of User Needs and Requirements and Operational Readiness

Once the system has been developed and all testing has been completed, the ST-CTN project team will conduct an ORT that will be documented in the ORTP. The objectives of these activities are to demonstrate that the deployed system operates as designed in a safe and secure manner before operational deployment begins. Operational readiness conceptually applies to the system itself as well as the implemented institutional and financial framework that supports, finances, and governs the deployed system.

Operational readiness will be established with a comprehensive set of tests and supporting demonstrations to be designed and conducted by the ST-CTN project team. A set of relevant tests will be conducted to verify that the system performs according to the documented System Requirements. The test results from these ORTs will be documented and reported to USDOT. Evidence of comprehensive and recursive testing conducted for the STP, conducted during the Agile Development Process, will be presented as evidence to verify operational readiness. Demonstrations are at a higher level and show that the system performs as expected in key use cases/scenarios. Demonstrations are differentiated from tests by the following general features:

- Exhibit a set of selected integrated, end-to-end system capabilities central to the deployment ConOps (e.g., key use cases); and
- Conducted as a live, real-time activity for the USDOT AO and AOR and federal team wherein success and failure of the demonstration are directly observable.

The ORTP will incorporate (at a minimum) the following elements for each test:

- *Test Descriptions*. Test Descriptions will include written descriptions of the individual V&V processes that will occur as part of the effort to ensure that the system was built correctly and that the correct system was built. Test descriptions will be linked back to documented System Requirement(s) whose fulfillment they will determine. The document will include a requirements-to-test procedures matrix that shows the test coverage relationship among the tests and the requirements. Every requirement will have at least one test case associated with it and each test case should have at least one requirement associated with it.
- *Test Cases*. Each test case will include a set of test inputs, execution conditions, and expected results developed for a particular objective, such as to exercise a particular path

within a system or a software application or to verify compliance with a specific requirement or set of requirements.

- *Test Procedures*. Test procedures spell out exactly how one verifies and validates that the component of the system examined actually functions as intended and as desired. Verification can use inspection, test, demonstration, and analysis but must be identified in each test. Each test procedure will include the trigger action(s), tester's action, and expected result.
- *Test Data*. Test Data will include scripts used to execute software operations, data that must be entered by someone as part of the process of V&V of the system and its component integration, or a description of what system-generated data will flow through different components of the system to accomplish a system function.
- *Test Results*. Documents that describe the results of each test conducted. The ORTP describe how test results will be summarized and documented across all tests and delivered to USDOT.
- *Test Failure Remediation*. This section will describe the actions to be taken in the event of a failed test. The ORTP will describe test criteria for pass/fail based on the parameters set in **Section 3.2.8**.
- *Schedule*. This section will describe the schedule for conducting the ORTs and any interdependencies between the different test cases.

The ORDP will incorporate (at a minimum) the following elements (adapted for the aspect of deployment readiness tested) for each demonstration:

- *Demonstration Descriptions*. The descriptions identify the objective, general location, participants, equipment, and actions to be taken within the demonstration to illustrate the successful deployment of key use cases.
- *Demonstration Procedures*. Procedures describe the sequence of events expected to be demonstrated and observable validation criteria associated with the overall purpose of the demonstration.
- *Demonstration Data*. Demonstration data will be collected before, during, or after the demonstration to support the observable demonstration validation criteria related to demonstration success (e.g., pass or fail).
- *Demonstration Results*. Documents that capture the results of each demonstration conducted. The ORDP will also describe how demonstration results will be summarized and documented across all demonstrations and delivered to USDOT.

3.2.10 Operations and Maintenance Processes

The Operations and Maintenance processes will be guided by the development of the CMOP and implemented in accordance with the SOMS. The CMOP will deal specifically with managing the technologies of the system. It will not include traveler training and customer operations. Those tasks will be covered by other documentation. In addition, a general section will cover operations and maintenance management.

The O&M Management section will cover the following areas:

- System components and software configurable items
- Software classification – this will drive the software maintenance levels and procedures described in the maintenance section.
- Roles and responsibilities for system and subsystem operations and maintenance
- Service Desk and Emergency Communications
- Asset Management including:
  - Application and data portfolio
  - Document configuration management
  - Beacon inventory / asset management
  - Change Management Process

The **Operations section** will address the following areas:

- Tools and activities to monitor the system and subsystems including normal operations, communications, security, and safety
- Activities to manage impacts to the system, both planned and unplanned. These include:
  - Identifying anomalies and recovery methods
  - Initiating defect and discrepancy management process
- Tools and activities to backup data
- Activities to manage the system configuration
- Tools and activities to report on service level agreement (SLA) criteria
- Periodic training for operations and maintenance staff

The **Maintenance section** will address the following areas:

- Software maintenance levels including preventative, corrective, and software enhancement
- Data maintenance plan describing staffing, schedule, tools, storage, and other resources needed to conduct data maintenance
- Data curation plans by data custodians: reviewed by the data governance committee
- Beacon maintenance plan including plan for preventive maintenance and replacement
- Software maintenance plan including the following processes and activities:
  - Defect management process
  - Preventive maintenance activities (patches and version updates)

- Corrective maintenance activities (version updates)
- System enhancement activities
- Format, approval and distribution of Software Maintenance Notes to users.

The draft CMOP will be developed in Phase 2 for implementation during the Phase 3 pilot. The final version will be developed, integrated with legacy operations/maintenance activities, and approved prior to commencing with the Post Phase 3 operations and management.

3.3 Post Phase 3 Processes

The Post Phase 3 Processes have yet to be determined, yet, during the pilot deployment, the IOOs managing and contributing to the program will begin to formalize the organizational structures established for project sustainability. These organizational structures will be described by the roles and responsibilities identified in the CMOP, Memorandum of Understanding (MOU), and institutional relationships.

Upon competition of Phase 2, the ST-CTN project team will transition the project so that it can operate as part of routine operational practice. Phase 3 will be deployed for a minimum of eighteen (18) months. The ST-CTN project team will draw upon and update relevant Phase 1 deliverables, in particular the Institutional, Partnership, and Financial Plan.

A Comprehensive Transition Plan (CTP) will be developed that identifies the concepts, applications, governance framework, agreements, key documents, and equipment to be maintained as elements of routine operational practice after the completion of Phase 3. The CTP will have one section for concepts and applications found to be successful and included in continuing operations, and one section for concepts and applications found to be unsuccessful and to be removed from continuing operations. The CTP will provide rationale for each successful and unsuccessful element. The CTP will also describe what organizational responsibilities will be taken in the post-deployment period compared to organizational responsibilities in Phase 3.

The CTP will also include documentation of the financial resources and agreements required to ensure financial sustainability in the post-deployment period for all continuing elements. Public and/or private sources of funds will be identified and if one or more new businesses are proposed to be a source of funds, a business plan containing standard elements shall be part of the documentation. Any dependencies on external organizations will be documented. The CTP will explicitly identify contingency plans with respect to identified uncertainties and other potential post-deployment issues posing a risk to successful post-deployment operations.

Phase 3 will operate in accordance with the Transition Plan. In addition, the ST-CTN system will also operate in accordance with the SMP operational approach to ensure safety of all users.

Appendix A. Acronyms and Glossary

This section includes a list of acronyms and a glossary of key terms used throughout the document.

Table 6. Acronyms

Acronym	Meaning
ADA	Americans with Disability Act
API	Application programming interface
ARC	Atlanta Regional Commission
ARC-IT	Architecture Reference for Cooperative & Intelligent Transportation
ATIS	Advanced Traveler Information System
ATL RIDES	Atlanta Rider Information and Data Evaluation System
CAD	Computer-aided dispatch
CAP	Comprehensive Acquisition Plan
CCL	Community Coordinator Lead
CDP	Connected data platform
CMOP	Comprehensive Maintenance and Operations Plan
ConOps	Concept of Operations
COR	Contracting Officer Representative
CTP	Comprehensive Transition Plan
CV	Connected vehicle
CV1K	Regional Connected Vehicle Infrastructure Deployment Program

Acronym	Meaning
C-V2X	Cellular Vehicle to Everything
CVTMP	Connected Vehicle Technology Master Plan
DL	Deployment Lead
DMP	Data Management Plan
DPM	Deputy Project Manager
DPP	Data Privacy Plan
DRSB	Deployment Readiness Summary Briefing
DSRC	Dedicated Short-Range Communication
EMT	Executive Management Plan
ETRA	Enabling Technology Readiness Assessment
FAQ	Frequently asked question
GCT	Gwinnett County Transit
GDOT	Georgia Department of Transportation
GTFS	General Transit Feed Specification
HLD	High-Level Design
ICTDP	Integrated Complete Trip Deployment Plan
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IOO	Infrastructure owner operator
IORS	Installation and Operational Readiness Schedule

Acronym	Meaning
IPFP	Institutional, Partnership and Financial Plan
ISO	International Organization for Standardization
ITS	Intelligent transportation system
IVR	Integrated voice response
LDL	Local Deployment Lead
LEP	Limited English proficiency
MAP	Map Data
MVP	Minimum viable product
OBU	On-board unit
OIS	Outreach Implementation Schedule
ORP	Operational Readiness Plan
ORT	Operational Readiness Test
ORTP	Operational Readiness Test Plan
OSS	Open-source software
OTP	Open Trip Planner
PED-SIG	Mobile Accessible Pedestrian Signal System
PML	Project Management Lead
PMP	Project Management Plan
PSM	Personal safety message
PTI	Pedestrian transit indication
QA	Quality assurance
RSU	Roadside unit

Acronym	Meaning
RTM	Requirements Traceability Matrix
SAD	Systems Architecture Document
SDD	Systems Design Document
SDLC	Software development lifecycle
SDS	Software Development Schedule
SEBOK	Systems Engineering Body of Knowledge
SEL	Systems Engineering Lead
SEMP	Systems Engineering Management Plan
SMP	Safety Management Plan
SMUG	Secure Mobile Unit Gateway
SOMS	System Operations and Maintenance Schedule
SPaT	Signal phasing and timing
ST-CTN	Safe Trips in a Connected Transportation Network
STM	Space time memory
STP	System Test Plan
SyRS	System Requirements Specifications
TSP	Transit signal priority
TSR	Transit stop request
UI	User interface
UNIRP	User Needs Identification and Requirements Planning
UX	User experience
V&V	Verification and Validation

This table includes a list of terms and definitions that are relevant for this document.

Table 7. Glossary

Term	Definition
Agile Software Development	Various processes to software development under which requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams and their customer(s)/end user(s). It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages rapid and flexible response to change. [Wikipedia]
Application programming interface (API)	Enables companies to make available the data of their products and services to external developers and business partners. This allows multiple services and products from different companies to communicate and leverage each other's data for improved collaboration, innovation, and added security [API].
Advanced Traveler Information System (ATIS)	A system that collects, aggregates, and disseminates transportation information, such as traffic, transit, weather, and connected vehicle data. This data is aggregate into data environments allowing for the dissemination of this information to travelers via mobile devices. [ATIS]
Burn-down Chart	An optional, but often used, indication of the amount of Product Backlog turned into an Increment of product during a Sprint by a Scrum Team, tracked by the Developers for use within the Scrum Team. [Scrum.org]
Computer-aided dispatch (CAD)	Software used to monitor transit operations and help with the management of transit operations. The software takes in information on transit routes, schedules, trip orders and vehicle assignments so that dispatchers are aware of their agency's transit vehicles' locations. [CAD]
Daily Scrum (or Stand-up)	Scrum Event that is held each day for the Development Team. At it, the Development Team plans work for the next 24 hours. This optimizes team collaboration and performance by inspecting the work since the last Daily Scrum and forecasting upcoming Sprint work. The Daily Scrum is held at the same time and place each day to reduce complexity. [Scrum.org]

Term	Definition
Dedicated Short-Range Communication (DSRC)	A protocol for communication between vehicles and between moving vehicles and fixed roadside access points. The protocol addresses safety critical issues associating with sending and receiving data. The protocol provides low-latency data-only V2V and V2I communications. [CAV]
Definition of Done	A formal description of the state of the Increment when it meets the quality measures required for the product. The moment a Product Backlog item meets the Definition of Done, an Increment is born. The Definition of Done creates transparency by providing everyone a shared understanding of what work was completed as part of the Increment. If a Product Backlog item does not meet the Definition of Done, it cannot be released or even presented at the Sprint Review. [Scrum.org]
Definition of Ready	Clear criteria that a user story must meet before being accepted into an upcoming iteration. [Agile Alliance]
Epic	A large user story, typically one that is too big to fit in a single Sprint. Epics need to be broken down into smaller user stories at some point before implementation as part of a Sprint. [Agile Alliance]
General Transit Feed Specification (GTFS)	A data specification that allows public transit agencies to publish their data to be consumed by a variety of transit-related applications. This data includes schedule, fare, and vehicle position which can be used to predict arrival times and display real-time information. [GTFS]
Minimum Viable Product (MVP)	A product with just those minimal features that allow it to be deployed, and no more.
Mobile Accessible Pedestrian Signal System (PED-SIG)	A mobile application system that exchanges information between roadside or intersection sensors and mobile devices carried by a pedestrian. The system is used to inform impaired pedestrians when to begin traversing a crosswalk and how to remain within the crosswalk. [CAV]
On-Board Unit (OBU)	An ITS related hardware that performs the data exchange between the infrastructure and a vehicle and installed in a vehicle (includes an after-market device). An OBU may contain applications that process the data received from the infrastructure and other sources such as another OBU. [CI]
Product Backlog	A Scrum Artifact that consists of an ordered list of the work to be done in order to create, maintain and sustain a product. [Scrum.org]

Term	Definition
Product Roadmap	A high-level plan that shows when in the future new products are expected to be developed or introduced by the organization/team. The requests to edit the road map (usually by adding new products) come from the sales force or senior management in the company when the marketing strategy is made. [ScrumBok]
Product Vision	A statement describing the desired future state that would be achieved by developing and deploying a product. A good product vision is simple, easy to understand statement and provides a coherent direction to the people who are asked to realize it. [ScrumBok]
Regression Testing	Testing following modifications to a test item or to its operational environment, to identify whether regression failures occur Note 1 to entry: The sufficiency of a set of regression test cases depends on the item under test and on the modifications to that item or its operational environment. [ISO/IEC/IEEE 29119-1-2013, §4.32]
Roadside Unit (RSU)	A transportation field device that performs the data exchange between OBUs, MUs, and other infrastructure elements. [CI]
Scrum	Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems. [Scrum.org]
Secure Mobile Unit Gateway (SMUG)	Serves as a secure means of exchanging information between a mobile unit (or proxy such as the ATL RIDES subsystem) and the CV environment. The MU Gateway serves to provide authenticated access, validate messages, transform and direct the messages to the appropriate destination. [ConOps]
Sprint	Scrum Event that is time-boxed to one month or less, that serves as a container for the other Scrum events and activities. Sprints are done consecutively, without intermediate gaps. [Scrum.org]
Sprint Backlog	Scrum Artifact that provides an overview of the development work to realize a Sprint’s goal, typically a forecast of functionality and the work needed to deliver that functionality. [Scrum.org]
Sprint Planning	Scrum Event that is time-boxed to 8 hours, or less, to start a Sprint. It serves for the Scrum Team to inspect the work from the Product Backlog that’s most valuable to be done next and design that work into Sprint Backlog. [Scrum.org]

Term	Definition
Sprint Retrospective	Scrum Event that is set to a time-box of 3 hours, or less, to end a Sprint. It serves for the Scrum Team to inspect the past Sprint and plan for improvements to be enacted during future Sprints. [Scrum.org]
Sprint Review	Scrum Event that is set to a time-boxed of 4 hours, or less, to conclude the development work of a Sprint. It serves for the Scrum Team and the stakeholders to inspect the Increment of product resulting from the Sprint, assess the impact of the work performed on overall progress toward the product goal and update the Product Backlog in order to maximize the value of the next period. [Scrum.org]
Test Case	Set of test case preconditions, inputs (including actions, where applicable), and expected results, developed to drive the execution of a test item to meet test objectives, including correct implementation, error identification, checking quality, and other valued information. [ISO/IEC/IEEE 29119-1-2013, §4.48]
Test Plan	Detailed description of test objectives to be achieved and the means and schedule for achieving them, organized to coordinate testing activities for some test item or set of test items. [ISO/IEC/IEEE 29119-1-2013, §4.75]
Test Procedure	Sequence of test cases in execution order, and any associated actions that may be required to set up the initial preconditions and any wrap up activities post execution. [ISO/IEC/IEEE 29119-1-2013, §4.78]
Test Script	Test procedure specification for manual or automated testing. [ISO/IEC/IEEE 29119-1-2013, §4.83]
Transit Signal Priority (TSP)	A part of a signal system that allows transit agencies to manage service by prioritizes buses and granting their right of way based on schedule adherence or passenger loads. [CAV]
User Stories	A statement (or a group of statements) that expresses the desired end user functionality. User Stories are generally simple, short, and easy to implement. Longer User Stories are further broken down into multiple User Stories. [ScrumBok]
Velocity	An optional, but often used, indication of the amount of Product Backlog turned into an Increment of product during a Sprint by a Scrum Team, tracked by the Development Team for use within the Scrum Team. [Scrum.org]

Appendix B. References

This section includes a list of documents referenced during the plan, including URLs and USDOT Publication Numbers, where possible.

Table 8. References

ID	Referenced Documents
[CVTMP]	AECOM. "Gwinnett County Connected Vehicle Technology Master Plan (CVTMP)." Duluth: Gwinnett County Department of Transportation. (2019). https://www.gwinnettcounty.com/static/departments/transportation/pdf/CVTechnologyMasterPlan2019.pdf
[UNIRP]	Atlanta Regional Commission. Deliverable Task 1B User Needs Identification and Requirements Planning (FHWA-JPO-21-852). Atlanta: U.S. Department of Transportation. (2021). https://rosap.ntl.bts.gov/view/dot/57010
[ConOps]	Atlanta Regional Commission. Deliverable Task 2 Concept of Operations (FHWA-JPO-21-857). Atlanta: U.S Department of Transportation. (2021).
[SyRS]	Atlanta Regional Commission. Deliverable Task 6 System Requirements Specifications (FHWA-JPO-21-880). Atlanta: U.S Department of Transportation. (2021).
[SC]	City of Columbus. Systems Engineering Management Plan for the Smart Columbus Demonstration Program (FHWA-JPO-17-518). Columbus: U.S. Department of Transportation. (2018). https://rosap.ntl.bts.gov/view/dot/34764
[ATIS]	Connected Vehicle Reference Implementation Architecture (CVRIA). Advanced Traveler Information System. Santa Ana: Iteris. (2016). https://local.iteris.com/cvria/html/applications/app4.html
[GTFS]	GTFS. General Transit Feed Specification Reference. Washington D.C.: GTFS. (2019). https://gtfs.org/reference/realtime/v2/
[API]	IBM Cloud Education. Application Programming Interface (API). Armonk: IBM. (2020). https://www.ibm.com/cloud/learn/api
[CI]	ICF, Wyoming Department of Transportation. Connected Intersection – Concept of Operations. Cheyenne: USDOT (2018). https://rosap.ntl.bts.gov/view/dot/41917

ID	Referenced Documents
[CAD]	Intelligent Transportation Systems, Joint Program Office. ITS Transit Fact Sheets – Technology Overview. Washington D.C.: U.S. Department of Transportation. (2021). https://www.pcb.its.dot.gov/factSheets/default.aspx
[NOBLIS]	Noblis, Inc. (2017). Applying Scrum Methods to ITS Projects. Washington D.C.: U.S. Department of Transportation.
[CAV]	Park, Hyungjun; Khattak, Zulqarnain; Smith, Brian. Glossary of Connected and Automated Vehicle Terms *Version 1.0*. Charlottesville.: University of Virginia Center for Transportation Studies. (2018). http://www.ctb.virginia.gov/resources/2018/oct/tech/glossary-of-cav-terms-ver1.0-03052018-1.pdf
[BAA]	U.S. Department of Transportation, Federal Highway Administration. (2020). ITS4US Broad Agency Announcement. Washington D.C.: U.S. Department of Transportation.

Appendix C. Source Code Management Plan

The ST-CTN project will support multiple open source code storage repositories, all hosted in GitHUB. The ATL RIDES open source code GitHub site is described in Section C-1 ATL RIDES and STM and the PMD open source code GitHub site is described in Section C-2 STM. Interfaces with other external subsystems will be hosted in one of the source code sites.

C-1 ATL RIDES

The ATL RIDES application is composed of several components. The routing engine (OpenTripPlanner) is open source and uses open-data specifications as inputs (GTFS, GTFS-realtime, and GTFS-flex). OpenTripPlanner is a multimodal trip planner launched by TriMet in 2009 that has since seen contributions from around the world. The ATL RIDES version (IBI fork) uses open-source TRANSIT-data-tools platform providing GTFS editing, deployment, and validation capabilities. In addition to IBI developers, third-party developers continually maintain and improve the open source code for these projects. Documentation, source code, open issues and other information for these projects can be found at the following repositories.

OTP Routing Engine and API: The primary application that ingests transit, street network, and shared-use mobility data sources and provides routing directions.

- GitHub (IBI fork)
- Documentation
- API docs

OTP React Redux: A front end library for writing modern OpenTripPlanner-compatible multimodal journey planning applications using React and Redux.

- OTP React-Redux GitHub
- OTP Middleware: Proxies requests from OTP UI to API, enhancing OpenTripPlanner with user storage, real-time trip monitoring, and more. GitHub.
- OTP Admin Dashboard: Dashboard application to manage OTP and otp-middleware. GitHub.

TRANSIT-data-tools: This application provides GTFS editing, management, validation, and deployment to OTP.

- GitHub (front end)
- GitHub (back end)
- Documentation

A key benefit of using open-source software is that any updates or feature releases to OpenTripPlanner made for a single client will automatically become available to other users of the platform. All users also benefit from bug fixes or other changes made for any particular OTP instance. The IBI developers also maintain an OTP component library (see https://www.opentripplanner.org/otp-ui/) that allows developers to easily integrate UI features into any website and customize them with their own styling and branding. An example of this is the next-generation TriMet trip planner (https://beta.trimet.org/).

C-2 STM

The STM application is open source (https://github.com/arpa-e-transnet/gtstm) and ingests data from multiple sources as identified in the DMP. A new GitHUB repository will be created for the updated STM used in this project. Documentation, source code, open issues and other information for these projects will be available at this repository. The repository will have separate folders for the following:

- Ingestion source code
- Backend source code
- Ingestion documentation
- Backend documentation
- API Documentation

The source code versioning will follow semantic versioning best practices Semantic Versioning 2.0.0. The version numbers will be in the *Major.Minor.Patch* format.

- **major version number** change will be used when there are incompatible API changes,
- **minor version number** for addition of backwards compatible functionality
- **patch update** for backwards compatible bug fixes.

U.S. Department of Transportation
ITS Joint Program Office-HOIT
1200 New Jersey Avenue, SE
Washington, DC 20590

Toll-Free "Help Line" 866-367-7487
www.its.dot.gov

FHWA-JPO-21-915

U.S. Department of Transportation