

# System Test Plan (STP)

## Buffalo, NY ITS4US Deployment Project

[www.its.dot.gov/index.htm](http://www.its.dot.gov/index.htm)

**Final Report — July 17, 2023**  
**FHWA-JPO-23-995**



U.S. Department of Transportation



Produced by Niagara Frontier Transportation Authority  
U.S. Department of Transportation  
Intelligent Transportation Systems Joint Program Office  
Federal Highway Administration  
Office of the Assistant Secretary for Research and Technology  
Federal Transit Administration

## Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The U.S. Government is not endorsing any manufacturers, products, or services cited herein and any trade name that may appear in the work has been included only because it is essential to the contents of the work.

---

## Technical Report Documentation Page

|   |   |  |                  |
|---|---|--|------------------|
| <b>1. Report No.</b><br><b>FHWA-JPO-23-995</b>  | <b>2. Government Accession No.</b>                          | <b>3. Recipient's Catalog No.</b>                            |                  |
| <b>4. Title and Subtitle</b><br>Phase 2 System Test Plan (STP)<br>Buffalo, NY ITS4US Deployment Project   |   | <b>5. Report Date</b><br>July 17, 2023                       |                  |
|   |   | <b>6. Performing Organization Code</b>                       |                  |
| <b>7. Author(s)</b><br>Polly Okunieff (ICF), Nayel Urena Serulle (ICF), Deepak Gopalakrishna (ICF), Robert Jones (NFTA), Kelly Dixon (GBNRTC), Jamie Hamann-Burney (BNMC), Darren Weibler (Neaera), Tony English (Neaera), Darlene Magold (Etch)  |   | <b>8. Performing Organization Report No.</b>                 |                  |
| <b>9. Performing Organization Name and Address</b><br>NFTA, 181 Ellicott Street, Buffalo, NY 14203<br>BNMC, 640 Ellicott Street, Buffalo, NY 14203<br>ICF International, 9300 Lee Highway, Fairfax, VA 22031<br>University at Buffalo, Amherst, NY 14228<br>RSG, 55 Railroad Row, Suite 101, White River Junction, VT 05001<br>ETCH, 4696 Smothers Road, Westerville, OH 43081  |   | <b>10. Work Unit No. (TRAIS)</b>                             |                  |
|   |   | <b>11. Contract or Grant No.</b><br>693JJ32250011            |                  |
| <b>12. Sponsoring Agency Name and Address</b><br>U.S. Department of Transportation<br>ITS Joint Program Office<br>1200 New Jersey Avenue, SE<br>Washington, DC 20590  |   | <b>13. Type of Report and Period Covered</b><br>Final Report |                  |
|   |   | <b>14. Sponsoring Agency Code</b><br>HOIT-1                  |                  |
| <b>15. Supplementary Notes</b><br>Elina Zlotchenko (USDOT ITS-JPO) is the Award Officer Representative (AOR) and Sarah Tarpgaard (USDOT) is the Award Officer (AO).   |   |  |                  |
| <b>16. Abstract</b><br><p>The Buffalo NY ITS4US Deployment Project seeks to improve mobility to, from and within the Buffalo Niagara Medical Campus by deploying new and advanced technologies with a focus on addressing existing mobility and accessibility challenges. Examples of the technologies to be deployed are electric and self-driving shuttles, a trip planning app that is customized for accessible travel, intersections that use tactile and mobile technologies to enable travelers with disabilities navigate intersections, and Smart Infrastructure to support outdoor and indoor wayfinding. The deployment geography includes the 120-acre Medical Campus and surrounding neighborhoods with a focus on three nearby neighborhoods (Allentown, Fruit Belt and Masten Park) with underserved populations (low income, vision impaired, deaf or hard of hearing, wheeled mobility device users and older adults).</p> <p>This document, the System Test Plan (STP), provides the system test strategy implemented for the proposed system. The STP describes the overall testing processes associated with each subsystem, and the integration of subsystems, external systems and applications as appropriate.</p> |   |  |                  |
| <b>17. Keywords</b><br>ITS4US; deployment; ITS; Intelligent Transportation Systems; system testing; accessibility   |   | <b>18. Distribution Statement</b>                            |                  |
| <b>19. Security Classif. (of this report)</b><br>Unclassified   | <b>20. Security Classif. (of this page)</b><br>Unclassified | <b>21. No. of Pages</b><br>66                                | <b>22. Price</b> |

# Revision History

| Name  | Date          | Version | Summary of Changes                         | Approver    |
|---|---------------|---------|--|-------------|
| Buffalo ITS4US Phase 2 System Test Plan (STP) | 14 March 2023 | 0.1     | Initial Draft                              | Kelly Dixon |
| Buffalo ITS4US Phase 2 System Test Plan (STP) | 16 June 2023  | 0.2     | Responded to high level USDOT Comments     |             |
| Buffalo ITS4US Phase 2 System Test Plan (STP) | 17 July 2023  | 1.0     | Changed cover page to reflect final status |             |

This page intentionally left blank.

# Table of Contents

|   |           |
|---|-----------|
| <b>1 Introduction.....</b>  | <b>1</b>  |
| 1.1 Scope.....  | 1         |
| 1.2 Intended Audience.....  | 1         |
| 1.3 Project Background.....   | 1         |
| 1.3.1 System Overview.....  | 3         |
| 1.3.2 System of Interest.....   | 6         |
| 1.4 Project Scope.....  | 14        |
| <b>2 System Test Plan Overview.....</b>                                   | <b>15</b> |
| 2.1 System Test Overview.....   | 15        |
| 2.1.1 System Test Phases and Processes Overview.....                      | 15        |
| 2.1.2 High Level Test Program Schedule.....                               | 17        |
| 2.2 Types of Testing.....   | 21        |
| 2.3 Test Identifiers.....   | 22        |
| 2.4 Verification Methods.....   | 22        |
| 2.5 Test Documentation.....   | 22        |
| <b>3 Agile Test Processes.....</b>  | <b>25</b> |
| 3.1 Agile Test Processes Overview.....                                    | 26        |
| 3.2 Unit and Integration Testing.....                                     | 27        |
| 3.3 Issues Tracking and Testing.....                                      | 27        |
| 3.4 Sprint Reviews and Demos.....   | 28        |
| <b>4 System Integration / Pre-production Testing.....</b>                 | <b>31</b> |
| 4.1 Integration Testing Overview.....                                     | 31        |
| 4.2 SDS Pre-production Testing.....                                       | 31        |
| 4.2.1 Inspection Testing.....   | 32        |
| 4.2.2 Demonstration.....  | 32        |
| 4.2.3 Field Testing.....  | 33        |
| 4.3 Smart Infrastructure System Integration / Pre-production Testing..... | 33        |
| 4.3.1 Inspection / Bench Testing.....                                     | 33        |
| 4.3.2 System Integration Testing.....                                     | 34        |

|  |           |
|--|-----------|
| 4.4 Full System Testing .....  | 36        |
| 4.5 System Dependencies .....  | 37        |
| <b>5 User Acceptance Testing .....</b>                               | <b>39</b> |
| 5.1 UAT Testing Overview.....  | 39        |
| 5.2 Reporting Defects .....  | 40        |
| <b>6 Defect / Deficiency Tracking and Resolution Processes .....</b> | <b>41</b> |
| 6.1 Defect / Deficiency Categorization and Impacts on Testing .....  | 41        |
| 6.2 Defect / Deficiency Resolution .....                             | 42        |
| 6.3 Defect / Deficiency Tracking and Reporting.....                  | 43        |
| 6.3.1 Defect Review Board.....                                       | 43        |
| 6.3.2 Defect Process and States .....                                | 43        |
| 6.3.3 Defect Tracking Sheet.....                                     | 44        |
| 6.3.4 Defect Reporting.....  | 46        |
| <b>7 Requirements and Test Case Traceability List.....</b>           | <b>47</b> |
| <b>Appendix A. References .....</b>                                  | <b>49</b> |
| <b>Appendix B. Acronyms.....</b>                                     | <b>51</b> |

## List of Tables

|   |    |
|---|----|
| Table 1. Demographics of Targeted Neighborhoods .....   | 3  |
| Table 2. Information Flows in System of Interest.....   | 7  |
| Table 3. High Level Test Program Schedule. ....         | 17 |
| Table 4. Definitions of Agile Methodology Terms.....    | 25 |
| Table 5. Ticket Types. ....                             | 28 |
| Table 6. System Integration Testing.....                | 35 |
| Table 7. System Level Defect/Deficiency Categories..... | 41 |
| Table 8. Testing Phases and Resolution Processes.....   | 42 |
| Table 9. Acronyms .....                                 | 51 |

## List of Figures

|   |    |
|---|----|
| Figure 1. Buffalo Niagara Medical Campus Relative to the Neighborhoods of Focus ..... | 2  |
| Figure 2. Proposed Service Area for the Community Shuttle.....                        | 5  |
| Figure 3. Community Shuttle Services .....  | 5  |
| Figure 4. System of Interest: High Level Context Diagram .....                        | 6  |
| Figure 5. Complete Trip Platform Subsystem.....                                       | 9  |
| Figure 6. Community Shuttle SDS System .....  | 10 |
| Figure 7. Community Shuttle HDS System Context Diagram .....                          | 11 |
| Figure 8. Indoor Navigation Module .....  | 12 |
| Figure 9. Pedestrian Signal Request (Ped-X) Module .....                              | 12 |
| Figure 10. Performance Measure Dashboard .....  | 13 |
| Figure 11. Components the Buffalo, NY ITS4US System .....                             | 14 |
| Figure 12. Testing Phases and Associated Testing Environments .....                   | 16 |
| Figure 13. Agile Process.....   | 29 |
| Figure 14. Defect Tracking Process. ....  | 45 |



# 1 Introduction

## 1.1 Scope

The System Test Plan (STP) defines the system test strategy, describing the flow of testing for software elements within the system testing process, including how the Agile process will include testing and user acceptance. In addition, the system inspection, integration and overall system acceptance testing is defined. A high-level overview of the each of these testing phases, along with the types of testing are described in this test plan. The STP includes discussions on the overall test processes associated with each subsystem, and integration of subsystems, external systems and applications as appropriate. In addition, the STP describes how test failures are handled; how regression testing is accomplished; how deficiencies/defects are handled and reported; and how the test results will be documented and stored.

## 1.2 Intended Audience

This document has been written to serve the information needs of multiple and varied audiences. This includes:

- USDOT - Intelligent Transportation System (ITS) Joint Program Office (JPO), Office of the Secretary (OST), Federal Highway Administration (FHWA), and Federal Transit Administration (FTA).
- Niagara Frontier Transportation Authority (NFTA): lead agency for deployment phases.
- NFTA project team members: ICF, University of Buffalo (UB), BNMC, Open Doors Organization, RSG (Resource Systems Group), and ETCH, as well as any independent evaluators or interested performance measurement and evaluation specialists.
- Public sector partners: NY State Department of Transportation, the Niagara Regional Transportation Commission, City of Buffalo, and NITTEC – a coalition of agencies across Western New York and Southern Ontario.
- Other important stakeholders: neighborhood associations, medical, healthcare, community, and human service organizations, and the BNMC Transportation Operations Council.

## 1.3 Project Background

Buffalo is moving toward a sustainable future for all, incorporating actions in the community, government, and private entities in the area. Providing access to the City's underserved populations to jobs and healthcare is the primary motivation for all the regional partners involved in this deployment. A lack of public transportation that adequately addresses "first/last mile" challenges is a major problem for community mobility, especially for people with disabilities. This often leads to compromised healthcare (e.g., rescheduled or missed appointments, delayed care) and/or dependence on paratransit service, which is much costlier for transit agencies and can be burdensome for riders. The ITS4US concept proposed here directly addresses these concerns by:

1. **Focusing on providing transit access to healthcare and jobs** to underserved residents or persons and allowing them to share in the economic development in downtown Buffalo.
2. **Putting technology to work in support of accessible transportation**, bringing leading edge researchers in accessible transportation, transit, and connected automation to solve a transportation need.
3. **Developing a scalable model** for considering accessibility and universal design in transportation technology projects.

The deployment location is targeted around the downtown Buffalo area with a focus on travel to and from the Buffalo Niagara Medical Campus (BNMC). The deployment includes the 120-acre Medical Campus and surrounding neighborhoods with a focus on three nearby neighborhoods (Fruit Belt, Masten Park, and Allentown)—see **Figure 1**.

More than 16,000 people work or study at the BNMC and more than 1.5 million visit each year for health care and other services, generating significant transportation demand for the area, its visitors, and its employees. The demographics of the surrounding neighborhoods (see **Table 1**) are emblematic of a broader socioeconomic and racial divide in Buffalo along Main Street, which this deployment seeks to bridge. In Allentown (west of Main Street), the percentage of traditionally underserved populations is significantly less than other neighborhood east of Main Street, namely Fruit Belt and Masten Park. Table 1 indicates percentages for Allentown that are far below average of the Metropolitan Statistical Area (MSA) in many categories, and percentages for Fruit Belt and Masten Park that are above average for the MSA.



**Figure 1. Buffalo Niagara Medical Campus Relative to the Neighborhoods of Focus**

*Source: Buffalo, NY ITS4US*

While the Allentown neighborhood is not characterized by underserved populations, it contains a high concentration of transit service and commercial activity, including health care offices. Allentown hosts several significant bus lines (including the #20-Elmwood, the #25-Delaware, the #11-Colvin, and the #8-Main) that connect the BNMC and Downtown Buffalo with neighborhoods to the north, carrying over 10,500 riders on an average weekday.

**Table 1. Demographics of Targeted Neighborhoods**

| Geography (ACS 2018 tracts) | Percent 0-veh. households | Percent population 65+ | Percent poverty | Percent Black | Percent Hispanic / Latino | Percent limited English ability | Percent income <\$25k | Percent with a disability (18 to 65 yrs old) | Percent veteran | Percent commute by transit | Total households | Total pop.     |
|-----------------------------|---------------------------|------------------------|-----------------|---------------|---------------------------|---------------------------------|-----------------------|--|-----------------|----------------------------|------------------|----------------|
| Fruit Belt                  | 47.0%                     | 21.9%                  | 28.0%           | 77.0%         | 8.9%                      | 4.2%                            | 39.5%                 | 20.0%  | 6.7%            | 16.1%                      | 976              | 2,435          |
| Allentown                   | 18.4%                     | 6.2%                   | 28.8%           | 7.2%          | 6.6%                      | 0.0%                            | 17.4%                 | 8.0%   | 7.8%            | 4.8%                       | 1978             | 3,143          |
| Masten Park                 | 35.0%                     | 18.5%                  | 34.7%           | 89.7%         | 3.1%                      | 2.9%                            | 38.9%                 | 15.2%  | 6.6%            | 11.7%                      | 1496             | 3,208          |
| Buffalo MSA                 | <b>36.6%</b>              | <b>12.0%</b>           | <b>31.1%</b>    | <b>36.6%</b>  | <b>11.6%</b>              | <b>4.8%</b>                     | <b>30.7%</b>          | <b>9.7%</b>                                  | <b>5.7%</b>     | <b>11.5%</b>               | <b>11,0701</b>   | <b>255,423</b> |

BNMC sits adjacent to the Fruit Belt neighborhood, which has a poverty rate of 28%, and 47% zero-car households. Several community and social services are found within the neighborhood, which is relatively close to the wider array of services and jobs offered in downtown Buffalo. Several bus lines serve the area, although headways are relatively infrequent, ranging between ½ hour and one hour. Access to dispersed jobs in the suburbs via public transportation tends to be difficult. Although accessible to the Fruit Belt residents, the Niagara Frontier Transportation Authority (NFTA) Metro Rail station is 0.25 – 0.75 miles away, a distance that becomes amplified during the winter and for travelers with physical difficulties. While BNMC continues to improve pedestrian accessibility, sidewalk quality and intersection crossings still are a challenge for wheelchair users and users with audible or visual impairments. The Fruit Belt struggles with aging infrastructure and infrastructure management issues, issues that have been consistently noted in community forums over the years.

This project seeks to improve transportation access for this population and utilize an innovative approach to support effective trip-making. BNMC’s user population makes it an ideal location to test accessibility features for safety and usability. The ITS4US Buffalo project focuses on two primary trip purposes: employee-related travel and patient/visitor travel to the campus from the three adjacent neighborhoods.

### 1.3.1 System Overview

The Greater Buffalo-Niagara Regional Transportation Council (GBNRTC) established its vision for the region for 2050 in its “Moving Forward 2050 – A Regional Transportation Plan for Buffalo Niagara” (GBNRTC; University at Buffalo Regional Institute, The SUNY at Buffalo School of Architecture and Planning; Cambridge Systematics; TyLin International, 2018).

The project seeks to guide transportation investments to:

1. Raise the region’s standard of living
2. Support efficient freight movement
3. Maximize infrastructure resiliency
4. Support focused growth in communities (urban, suburban, and rural)
5. Ensure access to opportunities and services

6. Support healthy and safe communities through targeted transportation investment
7. Strengthen the fiscal health of local governments
8. Preserve and protect a healthy environment and accessible open spaces and waterways
9. Create a fully integrated and seamless transportation environment

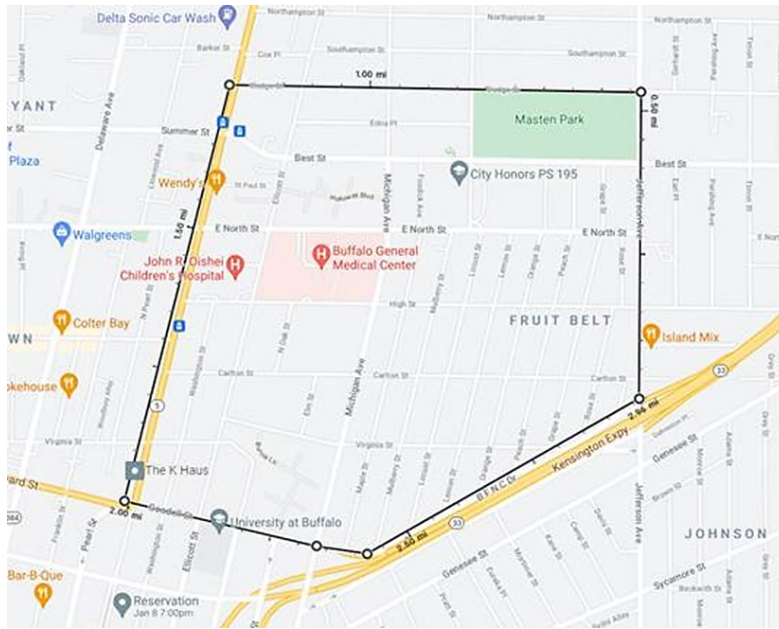
The Buffalo ITS4US project goals directly align with GBNRTC's goals 1, 4, 5, 6, and 9 by providing innovative tools and services to better enable travelers to make complete trips in and around the BNMC. Furthermore, the proposed system focuses on providing transit access to healthcare and jobs to underserved communities which will allow them to share in the economic development in downtown Buffalo.

To achieve these goals, the proposed system of interest is made of four major subsystems and a variety of data interfaces between them. The four major subsystems include:

- **Complete Trips Platform** – The complete trip platform (CTP) is the integrated trip planning function for travelers. It includes various modules that allow users to personalize their trip planning, execution, and navigation experience. Specific modules in this subsystem include:
  - User Profiles
  - Trip Booking
  - Trip Planning
  - Trip Monitoring and Notifications
  - Geolocation and Mapping
  - Navigation
  - Real-time situational monitoring
  - Performance metrics
  - Trip history/ledger
  - User Interface (UI): Mobile application
  - UI: Web
- **Community Shuttle Subsystem** – The Community Shuttle (CS) subsystem provides demand-responsive transit services within a specified zone of operations, using a mix of vehicles, including both human-driven (HDS) and self-driving shuttles (SDS). The SDS will operate on a predefined route(s), consisting of a set of streets within the zone and pick-up and drop-off locations, but will be responsive to travelers' demand (e.g., it can skip certain pick-up/drop-off locations if there is no demand). The HDS will provide door-to-door on demand service within the zone of operation. Modules within this subsystem include both types of vehicles, as well as a Shuttle Operations Center (SOC).
- **Smart Infrastructure Subsystem** – The smart infrastructure subsystem includes wayfinding and orientation for indoor and outdoor, provision of navigation and destination finding through information kiosks (Transportation Information Hub, TIH), augmented communications technologies (Smart Signs), and intersection treatment for hands-free, pedestrian signal requests.
- **Performance Dashboard Subsystem** – The performance measurement dashboard (PMD) subsystem measures and presents the performance of the system to the agency operating the system.

While not directly part of the project, the CS will be complemented by NFTA Paratransit Access Line (PAL) spontaneous (i.e., same day trip booking and execution) and regular services (i.e., trip reservation done by at least 8pm ET the day before the trip). Other NFTA services, such as bus and rail, will also provide complement to the CS.

The envisioned service area for the proposed CS fleet is shown in **Figure 2**. The services to be provided within this area are detailed in **Figure 3**.



**Figure 2. Proposed Service Area for the Community Shuttle**

Source: Buffalo, NY ITS4US

|                        |   | Services           | Inside Area   | Outside Area |
|------------------------|---|--------------------|---|--------------|
| Community Shuttle      | } | HDS                | 🚶 🚶   |              |
|                        |   | SDS                | 🚶 🚶   |              |
|                        |   | PAL (spontaneous)* | 🚶   |              |
|                        |   | PAL (regular)**    | 🚶   | 🚶            |
|                        |   | Other NFTA Serv.   | 🚶 🚶   | 🚶 🚶          |
| CTP - PAL eligible     |   | 🚶                  |   |              |
| CTP - General Traveler |   | 🚶                  |   |              |
|                        |   |                    | * PAL (spontaneous) refers to <u>same-day</u> service                 |              |
|                        |   |                    | **PAL (regular) reservation must be done by 8pm night before the trip |              |

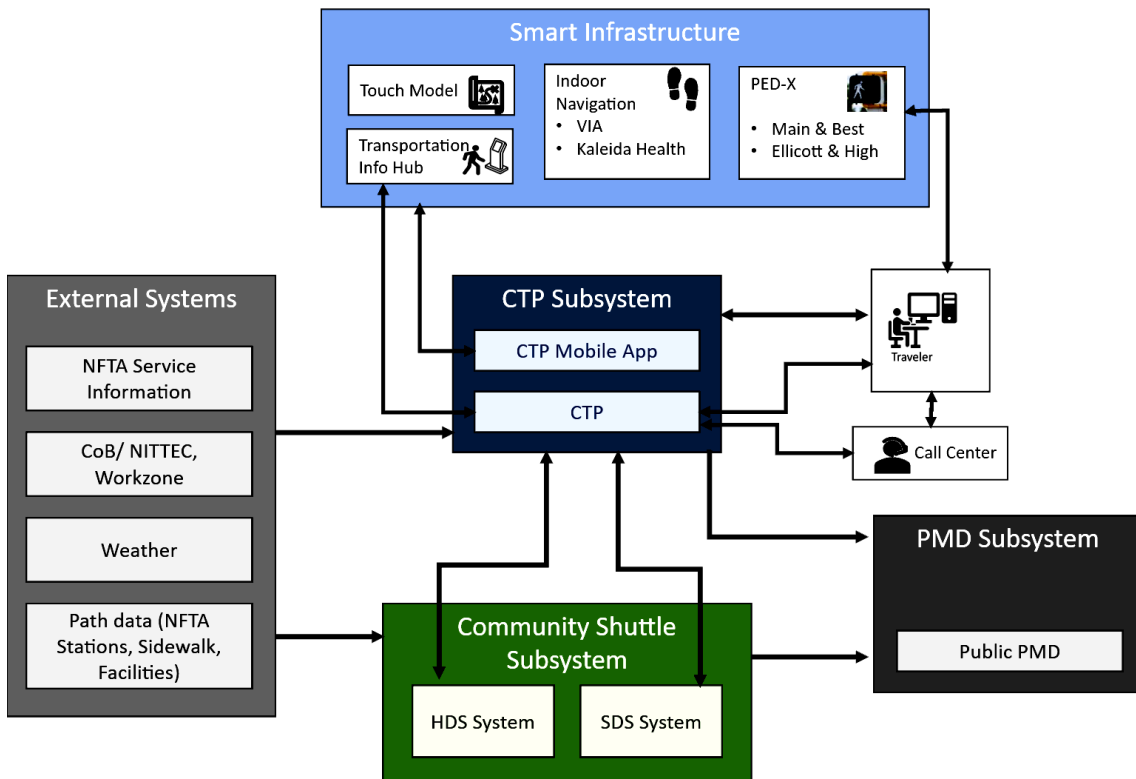
**Figure 3. Community Shuttle Services**

Source: Buffalo, NY ITS4US

### 1.3.2 System of Interest

This section describes a high-level description of the four subsystems of the Buffalo, NY ITS4US system. The following SOI diagrams are updated versions of the Context Diagrams described in the Concept of Operations (ConOps) (FHWA-JPO-21-860) and System Requirements Specification (SyRS) (FHWA-JPO-21-883).

The high-level context diagram which includes the four subsystems (Smart Infrastructure, CTP, CS, and PMD) are shown in **Figure 4**.



**Figure 4. System of Interest: High Level Context Diagram**

Source: Buffalo, NY ITS4US

Some of the details in the high-level context diagram were refined from the Phase 1 system of interest to Phase 2. Justification for these changes will be documented in the Phase 2 Concept of Operations, which will be available prior to June 14, 2024. The detailed context diagrams for each subsystem and associated Smart Infrastructure modules are described in the following four sections (each corresponding to one of the four subsystems). Where detail is needed, for example, the CTP, the subsystem functions, and flows are well documented. Where subsystems are deployed as a service, pre-existing systems, or turn-key, fewer details are provided (e.g., SDS and HDS).

The following subsections describe the key components of the SOI. The detailed diagrams include information flows within and between subsystems and functions. A complete set of these information flows (tagged in the diagrams as [l-n] where n is a number between 1 and 23), is

listed in **Table 2**. The table includes the information flow index, name, direction between source and destination, and target source and destination subsystem or component.

**Table 2. Information Flows in System of Interest**

| Information Flow # | Information Flow Name                                       | Information Flow Description  |
|--------------------|---|---|
| I-1                | UI Mobile App APIs  | Application Programming Interfaces (APIs) and web services provisioned between the CTP mobile app and server  |
| I-2                | UI Web  | APIs and web services provisioned to a thin client such as the web or TIH   |
| I-3                | Service Information   | NFTA fixed route GTFS, GTFS-realtime  |
| I-4                | BNMC Facility Map Update                                    | BNMC indoor facility pathways   |
| I-5                | NFTA Map Update   | NFTA station pathways (GTFS-pathways)   |
| I-6                | (1) SDS Booking Transactions<br>(2) SDS Service Information | (1) Reservations, scheduling and status information to book a SDS service<br>(2) SDS Service information (GTFS-realtime)  |
| I-7                | (1) HDS Booking Transactions<br>(2) HDS Service Information | (1) Reservations, scheduling and status information to book a HDS service<br>(2) HDS Service information (GTFS-realtime)  |
| I-8                | Ped-X Request Transactions                                  | (1) Information exchange between the CTP mobile app and Ped-X gateway to request actuation of the pedestrian crossing<br>(2) Location of Ped-X enabled crossing |
| I-9                | BNMC Facility Waypoint Sensor Broadcast                     | Transactions between mobile app native comm and indoor navigation waypoint sensors (e.g., beacons)  |
| I-10               | CTP Operational Log   | Operational and performance data monitored and collected by the CTP   |
| I-11               | Ped-X Operational Log                                       | A summary of the requests and their results   |
| I-12               | SDS Operational Log   | Operational and performance information from SDS  |
| I-13               | HDS Operational Log   | Operational and performance information from HDS  |
| I-14               | Performance Measures  | Performance Measurement results   |
| I-15               | Map Services  | APIs and web services that present performance measures   |
| I-16               | Direct Access Data Files                                    | APIs / links to access public data  |
| I-17               | API Data (requires authentication)                          | APIs and web services that require authentication   |

| Information Flow # | Information Flow Name  | Information Flow Description   |
|--------------------|--|--|
| I-18               | External Data (NFTA Performance Data and OSM / path / map updates) | OSM / path / map updates: Map, sidewalk, indoor facility and asset data and updates from external sources<br><br>NFTA Performance Data: Summary data for fixed route and PAL on time performance and other performance metrics   |
| I-19               | NITTEC Traffic Information   | The static network data and dynamic information includes right of way (ROW) data feeds and situational awareness TRANSCOM data fusion engine SPATAL data feeds. The following data feeds are currently identified:<br><br>-- mobile maps<br><br>-- situational awareness information (incidents, work zones, planned events) |
| I-21               | Communications between support personnel                           | Communications to support travelers on the CS SDS  |
| I-22               | Signal Control Exchange  | Message from the Ped-X gateway to a local traffic signal controller. The information flow forwards a request made by a traveler to request signal actuation  |
| I-23               | CoB PROW WZ  | Update of work zone information associated with the public right of way developed and disseminated by the City of Buffalo  |

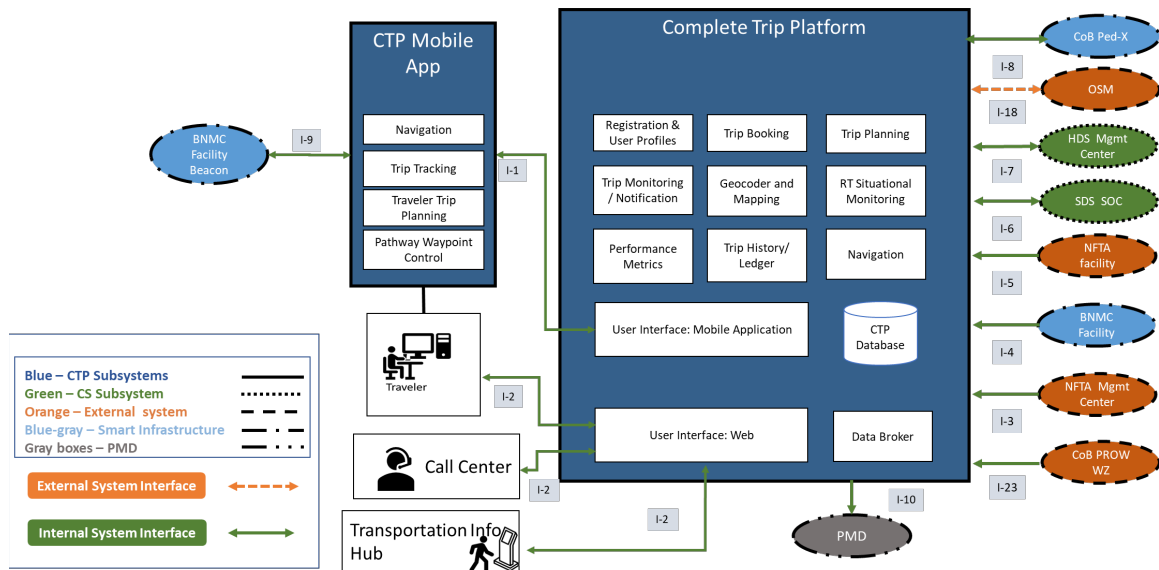
### 1.3.2.1 Complete Trip Platform

The CTP provides trip planning and travel functions for travelers. The tool is available for registered and non-registered account users. Account holders will be able to interact (e.g., book a trip reservation, check estimated time of arrival, etc.) with other mobility partners for which they have accounts (e.g., NFTA paratransit and community shuttle services), personalize their trip preferences and customize hands-free turn-by-turn notifications, and access wayfinding assets using components specified in the smart infrastructure subsystem. Non-registered travelers will be able to use the trip planning and travel tools to view accessible paths, transit services and alerts about asset status (e.g., elevator / escalator operations). The functions are described in the following sections.

The context diagram for the CTP is shown in **Figure 5**, where:

- 1) the subsystems are shown in blue boxes and boxes with icons;
- 2) functions are depicted in white boxes contained in the blue boxes;
- 3) terminators which are source or destinations (internal or external subsystems/systems) of the data are shown in various colored boxes and ovals as designated in the legend; and

- 4) information flows are shown as either green lines (designated as internal interfaces) or orange lines (designated as external interfaces) tagged with information flow indices [e.g., I-1]).



**Figure 5. Complete Trip Platform Subsystem**

Source: Buffalo, NY ITS4US

### 1.3.2.2 Community Shuttle Subsystem

The CS subsystem will provide on-demand mobility services within the Fruit Belt, Masten Park and Allentown neighborhoods and BNMC. Although they share similar interfaces to external components/subsystems, the CS consists of two service types – self driving and human driven services, each designated as a subsystem within this architecture description. The two subsystems include:

- **SDS System** is composed of the autonomous vehicle and the shuttle operations center (SOC). The SDS will provide microtransit service on a pre-defined set of road segments that satisfy the Operational Design Domain (ODD) of the SDS.
- **HDS System** is composed of wheelchair accessible vehicles, dispatch, and scheduling software (from the NFTA paratransit fleet) and reservations services from the NFTA PAL Direct software system. The HDS will provide door-to-door, on-demand service similar to current PAL paratransit service, but available for all registered CTP participants.

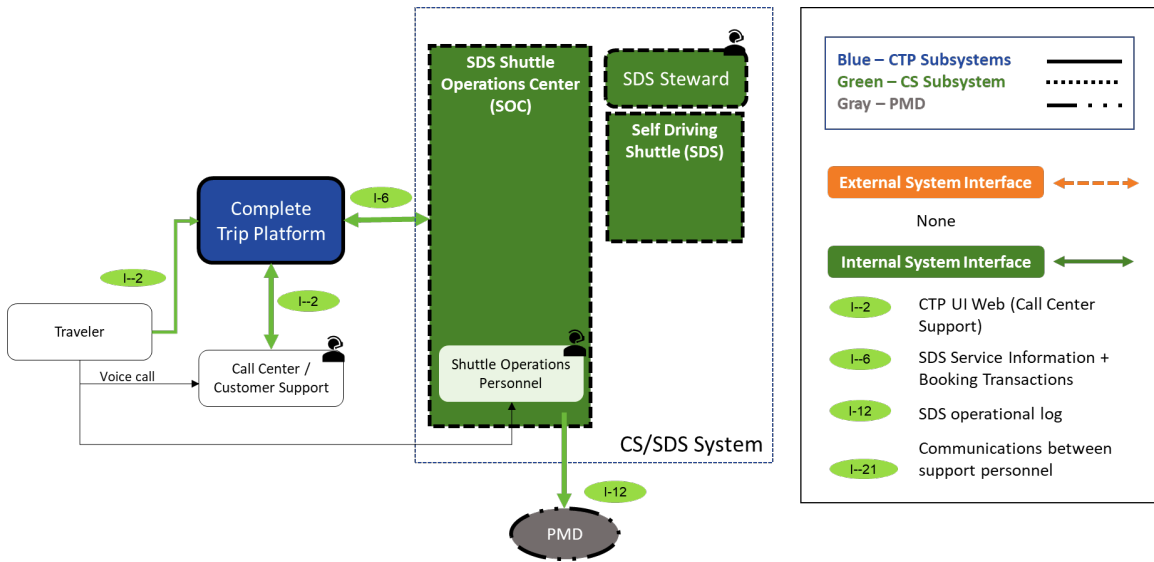
#### 1.3.2.2.1 The Self-Driving Shuttle (SDS) System

The SDS system will be procured as a turn-key service exchanging prescribed information flows between the SDS services and Buffalo project subsystems (e.g., PMD, CTP) and external systems (e.g., NITTEC). The SDS is composed of the Autonomous Vehicle and Shuttle Operations Center (SOC) components. The shuttle will operate on an on-demand schedule constrained to travel over a pre-defined route (i.e., a set of streets that satisfy the SDS ODD) and pre-designated pick-up/drop-off locations. The SDS SOC will receive all calls for services and will track the status of each vehicle in the SDS fleet. The SDS system will be procured as a turn-key

service exchanging prescribed information flows between the SDS services and SOI subsystems (e.g., PMD, CTP) and external systems (e.g., NITTEC).

Human actors include the SDS’s Shuttle Operations personnel who will manage the SOC, manage incidents, provide Call Center Customer Support (who provide direct support to travelers) and provide SDS operators / stewards (who will be trained by the SDS vendor).

**Figure 6** provides a representation of the SDS information flows with other systems and subsystems. Internal system interfaces are indicated by green lines and oval identification numbers and external system interfaces are represented with orange lines and oval identification numbers. Black lines represent voice calls. The CS/SDS System is enclosed within a dark blue box.



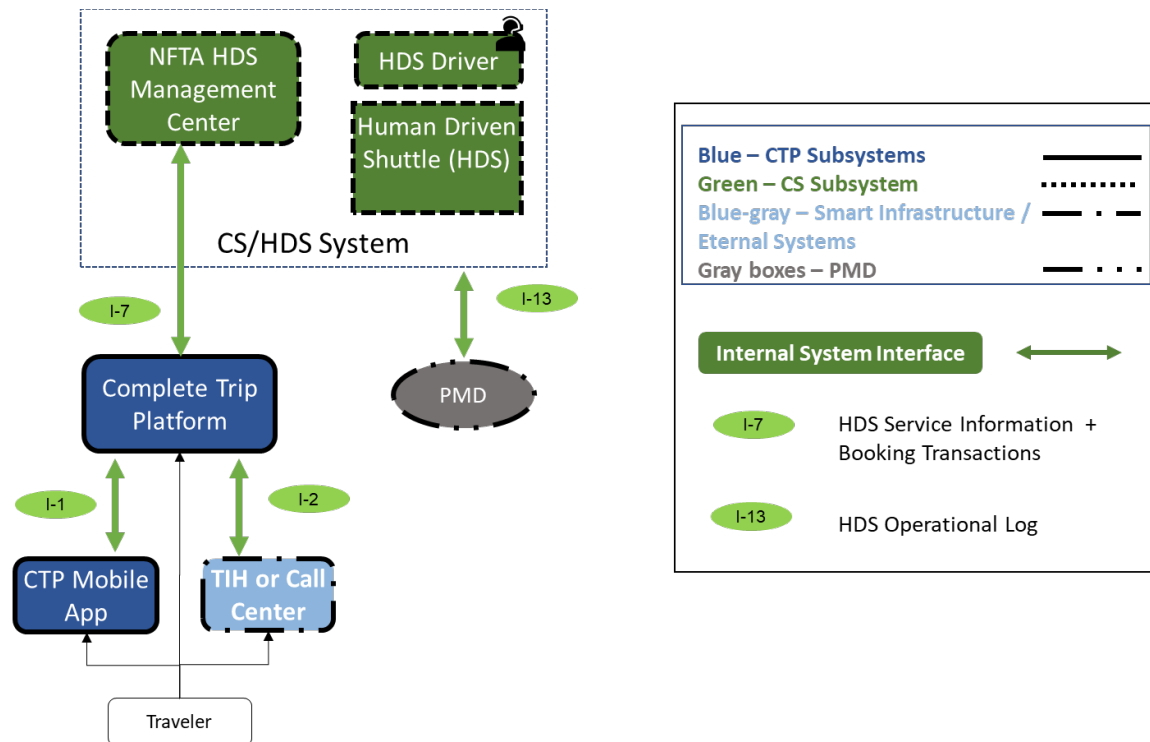
**Figure 6. Community Shuttle SDS System**

Source: Buffalo, NY ITS4US

### 1.3.2.2.2 The Human-Driven Shuttle (HDS) System

The HDS System (**Figure 7**) is composed of the NFTA Dispatch and Reservations System, HDS vehicle, and HDS operator. The HDS will use the NFTA PAL dispatch and software services and operate from the NFTA PAL dispatch and operations center. To that end, the major components are interfaces to and from the NFTA PAL Direct system.

The HDS will use an existing NFTA system that already provides the appropriate services (APIs) needed to transact customer booking and mobility services.



**Figure 7. Community Shuttle HDS System Context Diagram**

Source: Buffalo, NY ITS4US

### 1.3.2.3 Smart Infrastructure Subsystem

Smart Infrastructure (SI) supports personalized wayfinding capabilities for travelers. The technologies (modules) include:

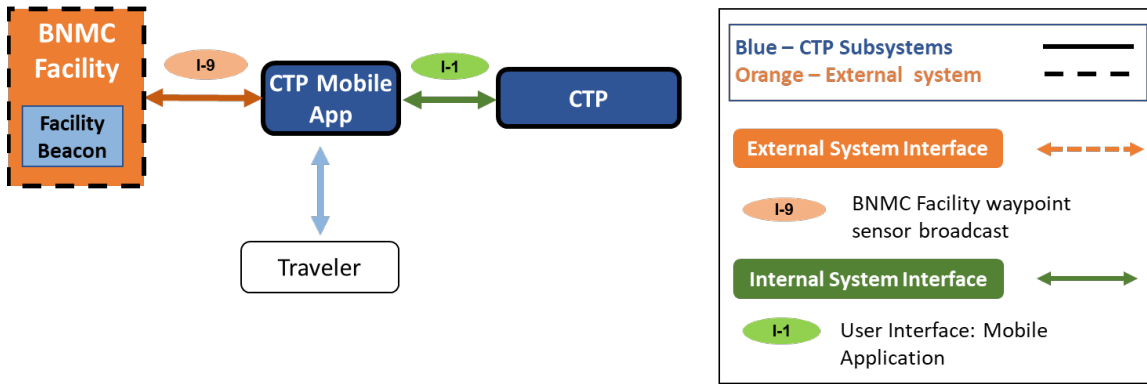
- Transportation Information Hub (TIH) modules
- Indoor Navigation modules
- Pedestrian Intersection Crossing (Ped-X) modules

SI provides support technologies for trip planning and wayfinding. The SI will be used for public access to trip planning activities, supplementary sensors to support indoor navigation at building / facilities and Metro rail stations, and broker / gateway services for pedestrians to request pedestrian actuation at selected signalized intersections—Main St. & Best St. and Ellicott St. & High St.

The context diagrams for the indoor navigation component and the pedestrian signal crossing component are provided below.

#### 1.3.2.3.1 Indoor Navigation Component

Using commonly available communications technologies already deployed in mobile handsets (i.e., smart phones), low-cost beacons will be deployed at two indoor spaces to provide indoor navigation (**Figure 8**). These beacons will provide waypoint (location) information for digital wayfinding integrated with the CTP mobile app navigation function.



**Figure 8. Indoor Navigation Module**

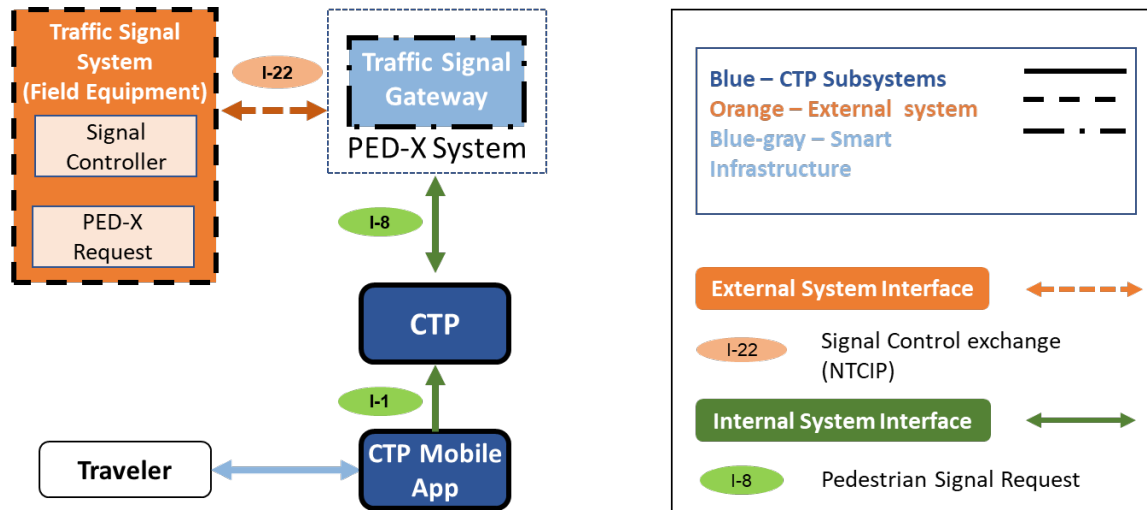
Source: Buffalo, NY ITS4US

1.3.2.3.2 Pedestrian Signal Crossing Module

The Ped-X module will be implemented as a Ped-X Gateway. This gateway receives a CTP generated message triggered by their trip plan. The gateway serves as a conduit to authenticate and secure information exchange between the CTP and traffic signal system to actuate the pedestrian request. Information channeled from the pedestrian to the signal system includes the request information.

Note that the Traffic Signal System, using an audible pedestrian signal will display and announce *walk* and *don't walk* signals at the intersection. This link is not shown in the figure because it is not a physical communications message.

The Pedestrian Signal Crossing module is shown in **Figure 9**.



**Figure 9. Pedestrian Signal Request (Ped-X) Module**

Source: Buffalo, NY ITS4US

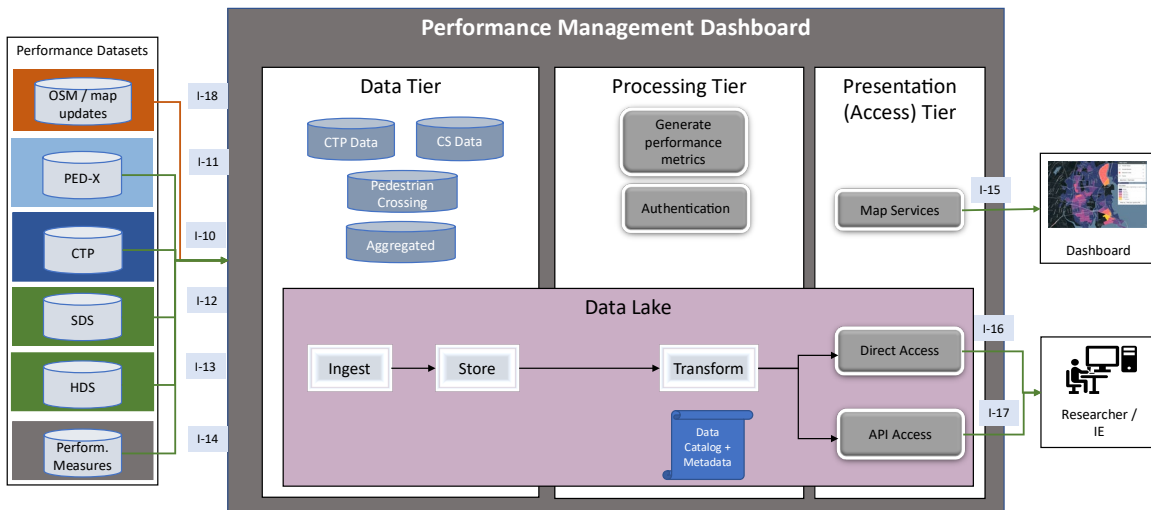
### 1.3.2.4 Performance Measure Dashboard Subsystem

The Performance Measure Dashboard subsystem monitors, integrates, analyzes, and displays performance measures from other subsystems and external sources. The subsystem includes functions to ingest log files from the subsystems and external data sources, storage, analytic and visualization tools to display and access current and historic data sets produced from the integrated system. The PMD subsystem will be implemented using a three-tier architecture – data (data tier), analytic processes (processing tier), and access / visualization (presentation tier), see **Figure 10**.

**Data Tier.** The data tier will ingest and store data from the other subsystems and external systems as needed (defined in the Phase 1 PMESP (FHWA-JPO-21-878), Phase 1 Data Management Plan (DMP) (FHWA-JPO-21-868)[1], and subsequent design documents). In addition, metadata management will be included to ensure data integrity as data is ingested and transformed for distribution. The data tier will ingest operational, maintenance, and performance summary data from each subsystem, as well as non-PII data from the performance measurement reports. The data will be ingested by the PMD and all PII will be removed.

**Processing Tier.** The processing tier will provide services to curate, transform, parse and query data stores to generate performance and aggregated measures. A user authentication function will provide access to users of different security levels. Security and privacy provisions will be implemented to protect, store and archive information.

**Presentation Tier.** The access and visualization channels will include a web-based dashboard showing key system performance measures as well as a data portal that will provide access via data feeds and APIs for authorized users.

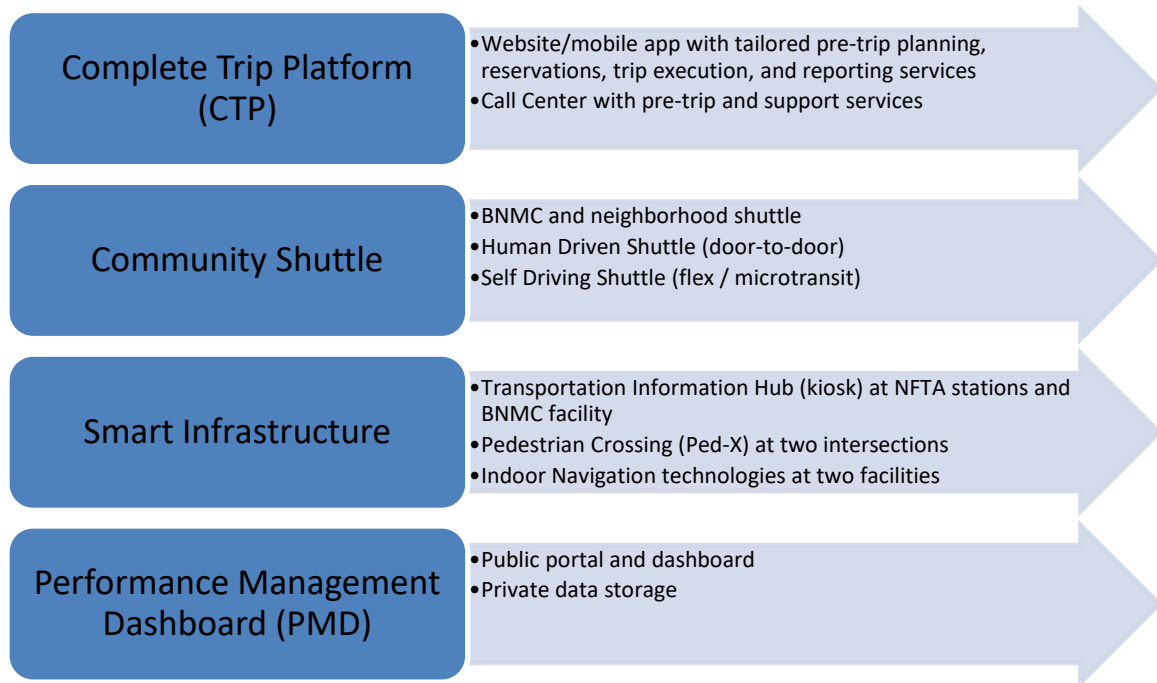


**Figure 10. Performance Measure Dashboard**

Source: Buffalo, NY ITS4US

## 1.4 Project Scope

The project seeks to deploy an integrated suite of technologies chosen to address identified needs of users and gaps within the systems and services provided. These technologies are represented by the four subsystems described in the previous subsections and include a Complete Trip Platform (CTP), Community Shuttle, Smart Infrastructure and Performance Measure Dashboard (PMD). **Figure 11** provides a summary of the subsystems and high-level scope.



**Figure 11. Components the Buffalo, NY ITS4US System**

*Source: Buffalo, NY ITS4US*

The Buffalo, NY ITS4US system was designed to address all the needs identified in the Concept of Operations (ConOps), satisfy all the requirements listed in the Systems Requirements (SyRS), and follow the architecture detailed in the System Architecture Design (SAD).

# 2 System Test Plan Overview

## 2.1 System Test Overview

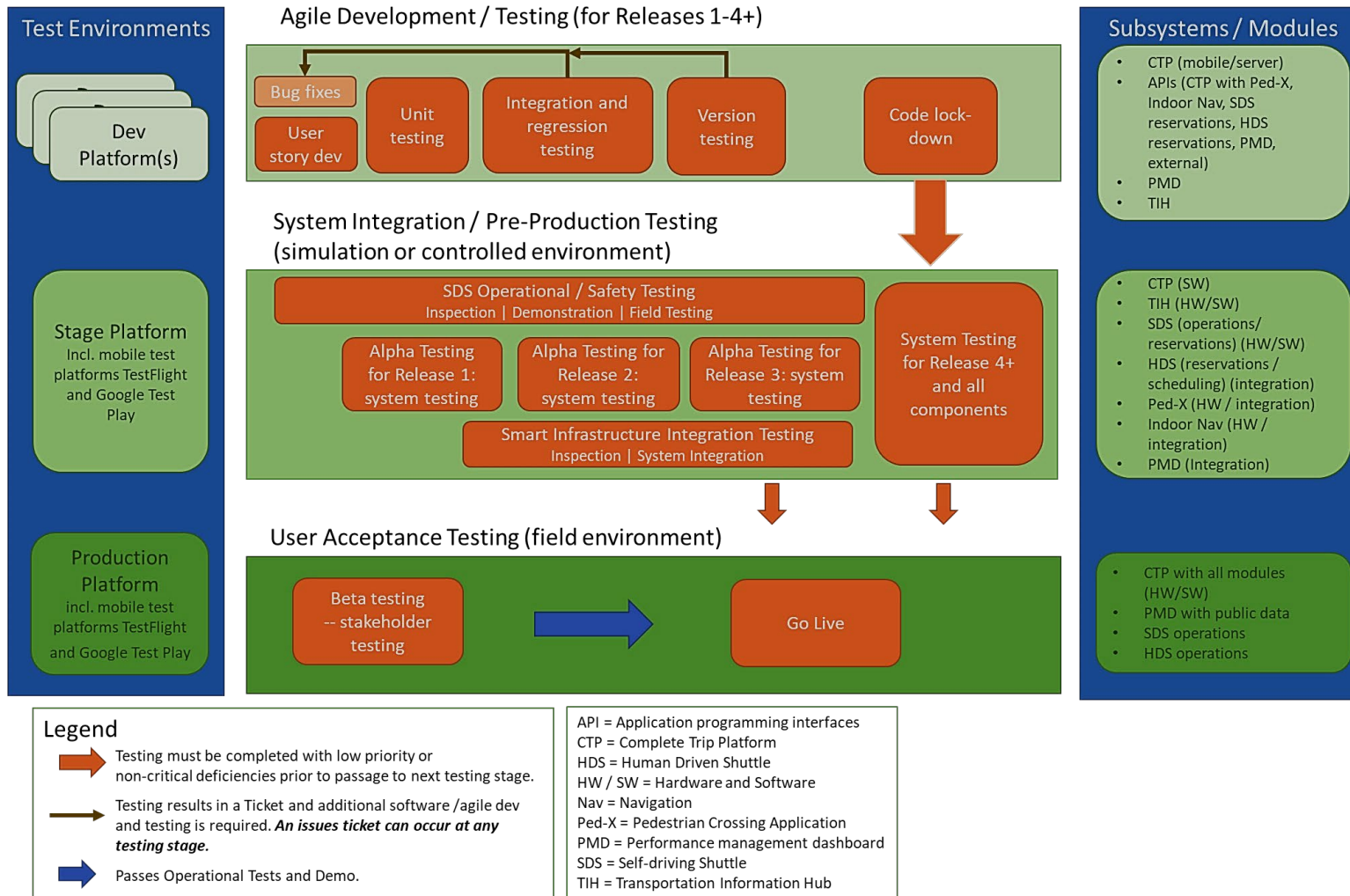
The System Test Plan (STP) details the testing procedures of all subsystems and system components of the Buffalo, NY ITS4US Deployment Project. The STP describes the system test phases and processes, the types of testing that each subsystem / application will include, the methods used to verify adherence to needs and requirements, and the documentation and artifacts that will be produced and stored through the testing processes.

### 2.1.1 System Test Phases and Processes Overview

The system test plan, shown in Figure 12, includes three major testing phases:

- **Agile Development and Testing** – unit and integration testing of software that is under development using an Agile methodology. The Agile development generates a pre-production staged version that is used for Alpha testing (in the pre-production testing phase). There are four releases that are progressed to the next phase.
- **System Integration / Pre-production Testing** – consists of two types of testing for the SDS, CTP releases and integration elements, and Smart Infrastructure:
  - Hardware inspection, demonstration and field testing of procured or turnkey systems including the TIH, Ped-X, indoor navigation (beacons) and SDS.
  - Controlled and formal pre-production testing of software and software integrated with service elements from hardware or software as a service application (e.g., indoor navigation, traffic management gateway, community shuttle operations). This includes formal testing of each CTP software releases.
- **User Acceptance Testing** – the final release will consist of Full System Integration Testing for all the system components including SDS, CTP, community shuttle scheduling, dispatch and operations, pedestrian crossing operations, indoor navigation, TIH, and performance measurement dashboard (analysis, presentation, and data access). Testing conducted by trained stakeholders and volunteers in a less formal way.

Each testing phase uses controlled Test Environments to apply safety requirements and capture anomalies and defects in the system. For defects, a ticket is issued to resolve and retest the component as well as a sample of related tests to verify the system continues to operate as expected. Components are listed in the Subsystems and/or Modules swimlane. The Test Environments, Testing Phases and Subsystems and Modules are identified in **Figure 12** and discussed in more detail in **Table 3** (Column *Description*).



**Figure 12. Testing Phases and Associated Testing Environments**  
(Source: Buffalo ITS4US)

## 2.1.2 High Level Test Program Schedule

This section provides a high-level test program schedule that shows approximately when each of the main testing phases will be accomplished.

**Table 3** provides a high level schedule for each testing phase by subsystem or system as depicted in **Figure 12**. The table includes the following:

**Test Phase** – the test phase by test type and subsystem.

**Description** – description of the test phase.

**Subsystem / Applications** – the components tested during the test phase. The components may include all the subsystem or applications or smaller systems within the subsystem.

**Release Date** – date range of the testing.

**Table 3. High Level Test Program Schedule.**

| Test Phase                | Description   | Subsystem/ Applications   | Release Dates  |
|---------------------------|---|---|--|
| Agile Testing for the CPT | <p>Agile testing includes three-weeks of development, test and integration cycles. Testing during each sprint includes unit and integration testing. Near the end of each release, version testing is conducted before the code is locked down and progressed to a staged, pre-production environment.</p> <p>The first three releases generated new code for the CTP software and integration with key applications (e.g., indoor navigation, reservations—both SDS and HDS, Ped-X and TIH).</p> <p>All software development is implemented using Agile Development Methodology.</p> | <p>CTP integrated with the following modules:</p> <ul style="list-style-type: none"> <li>Indoor Navigation APIs</li> <li>Reservations APIs for HDS, SDS and PAL</li> <li>Ped-X APIs</li> <li>TIH remote access</li> </ul> | <p>Release 1 (MVP): April 2023</p> <p>Release 2: October 2023</p> <p>Release 3: April 2024</p> <p>Release 4: June 2024</p> |
| Agile Testing for the PMD | <p>Agile testing includes three-week development, test and integration cycles. Testing during each sprint includes unit and integration testing. Near the end of each release, version testing</p>  | <p>PMD integrated with the following elements:</p> <ul style="list-style-type: none"> <li>CTP</li> </ul>  | <p>Release 1: October 2023</p>   |

| Test Phase  | Description  | Subsystem/ Applications  | Release Dates  |
|---|--|--|--|
|   | <p>is conducted before the code is locked down and progressed to a staged, pre-production environment.</p> <p>There are three releases scheduled for the PMD that are coincident with the CTP. This streamlines the pre-production testing phases so that CTP output could feed and verify the PMD integration, operations and performance.</p>  | <ul style="list-style-type: none"> <li>Community shuttle</li> <li>Ingestion from external sources</li> <li>Outputs to external sources including Independent Evaluators</li> </ul>   | <p>Release 2:<br/>April 2024</p> <p>Release 3:<br/>June 2024</p> |
| <p>System Integration / Pre-production Testing for the SDS</p>                                    | <p>System integration / pre-production testing for the SDS includes:</p> <ul style="list-style-type: none"> <li><b>Inspection</b> (e.g., Bench Testing) of the SDS components including operations center software and vehicles.</li> <li><b>Demonstration</b> of the performance and safety of the operations software and vehicles.</li> <li><b>Field Testing</b> in the UB proving grounds and BNMC designated travel path and pickup locations. In addition, system integration testing will be conducted to verify and validate reservations APIs are exchanged between the CTP and SDS.</li> </ul> | <p>SDS without passengers:</p> <ul style="list-style-type: none"> <li>Tactical or maneuvering behavior</li> <li>Operational Design Domain</li> <li>Object detection and collision avoidance</li> <li>Fail mode behavior</li> </ul> | <p>Release 5 onwards, starting on Jan 2025</p>                   |
| <p>System Integration / Pre-product Testing for the HDS Reservation-Scheduling-Dispatch (RSD)</p> | <p>System integration / pre-production testing for the HDS RSD includes:</p> <ul style="list-style-type: none"> <li>Demonstration of the performance and safety of the operations software and vehicles (the dispatch and operator modules).</li> <li>As part of <b>Release 2 system integration testing</b>, the CTP was integrated with the HDS RSD and controlled testing was conducted to test every function of the integrated and configured functionality.</li> </ul>   | <p>RSD:</p> <ul style="list-style-type: none"> <li>Demonstration of dispatch and operator modules</li> <li>System integration with CTP reservations (including end-to-end testing)</li> </ul>                                      | <p>April 2024</p>  |

| Test Phase  | Description   | Subsystem/ Applications   | Release Dates |
|---|---|---|---------------|
| System Integration / Pre-production Testing for the TIH               | <p>System integration / pre-production testing for the TIH includes:</p> <ul style="list-style-type: none"> <li>• <b>Inspection</b> of the TIH equipment to inspect the hardware and software components.</li> <li>• As part of <b>Release 3 system integration testing</b>, the CTP was integrated with the TIH and controlled testing (also called alpha testing) was conducted to test every function of the integrated and configured functionality.</li> </ul>   | <p>TIH:</p> <ul style="list-style-type: none"> <li>• Equipment inspection (bench testing)</li> <li>• Installation and activation</li> <li>• System integration with CTP</li> </ul>  | April 2024    |
| System Integration / Pre-production Testing for the Ped-X             | <p>System integration / pre-production testing for the Ped-X includes:</p> <ul style="list-style-type: none"> <li>• <b>Inspection</b> of the Ped-X equipment to inspect the hardware and software components. The installation procedures review the configuration parameters, integration and monitoring of the Miovision equipment with the signal controller. These procedures are described in the Comprehensive Installation Plan (CIP) for the MioVision equipment. In addition, demonstration procedures review configuration of the MioVision cloud services.</li> <li>• As part of <b>Release 3 system integration testing</b>, the CTP was integrated with the Ped-X and controlled testing was conducted to test every function of the integrated and configured functionality.</li> </ul> | <p>Ped-X Cloud Services:</p> <ul style="list-style-type: none"> <li>• System integration with CTP</li> </ul> <p>Ped-X intersection equipment:</p> <ul style="list-style-type: none"> <li>• Equipment inspection (bench testing)</li> <li>• Installation and activation</li> <li>• System integration with signal system and cloud services</li> </ul> | April 2024    |
| System Integration / Pre-production Testing for the Indoor Navigation | <p>System integration / pre-production testing for the Indoor Navigation elements includes:</p> <ul style="list-style-type: none"> <li>• <b>Inspection</b> of the Bluetooth beacons to ensure that they work properly prior to installation in the facilities. Following the initial inspection, and during installation, the beacons were tested for proper operations and signal strength to ensure that the coverage is consistent among the beacons</li> </ul>  | <p>Indoor Navigation Beacons:</p> <ul style="list-style-type: none"> <li>• Equipment inspection (bench testing)</li> <li>• Installation and activation</li> </ul> <p>Indoor Navigation Cloud Services:</p>  | April 2024    |

| Test Phase  | Description  | Subsystem/ Applications  | Release Dates        |
|---|--|--|----------------------|
|   | <p>that were installed. This step is not necessary for Kaleida Health General Hospital since the facility is already outfitted with beacons.</p> <ul style="list-style-type: none"> <li>As part of <b>Release 3 system integration testing</b>, the CTP was integrated with the Indoor Navigation software development kit (SDK). The SDK was piloted with the CTP in two controlled environments before it is tested in the two facilities in BNMC.</li> </ul>    | <ul style="list-style-type: none"> <li>Beacon mapping (in cloud mapping services)</li> <li>System integration with CTP</li> </ul>  |                      |
| System Integration / Pre-production Testing for the full system integration | By the CTP fourth (4 <sup>th</sup> ) software release all the systems are operational. To that end, comprehensive system integration testing is conducted to ensure that all systems operate as expected, meet system requirements and demonstrate user and safety needs. System integration testing was conducted to ensure that the releases are integrated with other modules and applications and that testing is conducted on mobile and web user interfaces. | All systems and subsystems   | June 2024            |
| User Acceptance Testing (UAT)   | UAT (also called beta testing) was conducted by a select group of trained staff and volunteers to exercise the elements of the system that passed the pre-production phase.  | Started in May 2024, all the subsystems and applications were deployed including passenger testing the HDS community shuttle services—SDS will be tested started on Jan 2025 | May 2024 – July 2025 |

## 2.2 Types of Testing

Each subsystem / application will include different types of tests to ensure the component or module is inspected, observed, verified and validated against the requirements:

- **Functional Testing** focuses on validating that the element operates and performs according to user expectations. Functional testing may be achieved by using test scripts that meet user acceptance criteria, demonstrations, and informal user acceptance testing (UAT).
- **Performance Testing** focuses on demonstrating that the system is meeting user expectations. Performance testing may be achieved by using automated test scripts that meet user acceptance criteria, demonstrations, and/or informal UAT.
- **Security Testing** focuses on ensuring that the system secures user data and addresses cyber security concerns. Typically, security testing occurs by using penetration testing (pen testing), intrusion testing using data and interface anomalies, or other security testing methods.
- **Interface Testing** focuses on demonstrating that two or more subsystems, applications or modules interoperate using open architecture components such as APIs, SDKs or queries. Interface testing may be achieved with typical, outlier or end point testing.
- **Integration Testing** focuses on combining two or more subsystems, applications or modules that operate together in an enterprise environment. Integration testing occurs when a sample of previous developed unit and integration tests are repeated in a regression testing phase.
- **Inspection**, also called bench testing, focuses on reviewing and exercising equipment, off-the-shelf (OTS) or software as a service software (SaaS) to ensure that all the parts and pieces work, and the system meets the functional, performance and environmental requirements (e.g., equipment, OTS, SaaS acceptance criteria).
- **Demonstrations** focuses on showing the incremental functionality integrated into releases to stakeholders. The demonstration verifies and validates functionality and operations to key stakeholders by walking through each function showing the expected outcome. The demonstration is used to solicit feedback from stakeholders to support system developers and integrators in enriching the user experience.
- **User Acceptance Testing** exercises the system in a real world environment as used by actual users. Since no comprehensive testing can exercise all scenarios, the UAT provides actual users who exercise the system in unexpected ways. Trained users use the tool in a production environment prior to its release. It is intended to vet the system for defects and anomalies.
- **Regression testing** is performed against a sample set of requirements and related requirements to verify that recent code or system changes, defect fixes, or performance updates have not adversely affected existing features. The sample set is determined by the scope and modularization of the subsystem or application.

## 2.3 Test Identifiers

Formal testing that occurs during the Agile and Pre-production phases will use testing procedures (e.g., automated or manual scripts) and be assigned a unique identifier that traces to one or more requirements (and Agile user stories).

UAT tests will not be tracked because they will be testing through beta testing—volunteer users who use the tests in a real world environment. Regardless, reported bugs, anomalies or user feedback will be assigned a unique identifier to trace its occurrence, testing platform and other identifying information.

## 2.4 Verification Methods

To verify adherence to needs and requirements, the System Test Plan will use four verification methods that show how a requirement was met. These include:

- **Inspection** – Where the requirement is verified by visual inspection or review of software code.
- **Demonstration** – Where the requirement is verified by demonstrating a specific capability or action. Demonstration is usually verified during the execution of the test itself; you witness the capability accomplishing the requirement.
- **Analysis** – Where the requirement is verified by an analysis method. As an example, Mean Time Between Failure (MTBF) requirements are usually verified by analysis since it may not be feasible to wait months for a potential failure of a component.
- **Test** – Where the requirement is verified by a test that is run and then later analysis is conducted on the data collected during the test. Many times, performance requirements are verified by test as it may not be immediately obvious if a performance requirement was met until analyzing logs to determine the precise times when events occurred.

## 2.5 Test Documentation

This section lists the test documentation and artifacts that will be produced and stored. The documents include the following:

- **System Test Plan** – The test plan describes the testing processes for each of the system subsystems, applications and hardware elements.
- **User Story and Sprint Test Artifacts** – Many of the Agile user stories are tested through demonstration or automated scripts. The user stories include acceptance criteria. When the criteria are demonstrated, the sprint is documented as “done” and closed in Jira.
- **System Test Procedure** – A formal test procedure is developed and conducted during the pre-production phase for each major release or subsystem of the system. The test procedure describes one or more **test cases** that are manually conducted to exercise the

system. The procedures will enumerate the anomalies and outliers incorporated in the test cases and data to detect boundary issues and error conditions. The procedure includes information on the test platform/environment, conditions, restrictions, and input data used to perform the set of test cases.

- **System Test Cases** – Each test case includes an introduction, body and scoring section. The introduction describes the purpose, requirement(s) under test (if applicable), the execution conditions (i.e., trigger event), and inputs. The body of the test case includes a sequence of actions, expected results, remarks, and blank entry for the tester to mark the results. The end of each test case includes a scoring section for the tester to mark pass or identify any defect in the system.
- **System Test Case Results** – The system test procedures will be administered by one or more testers in a field or simulated environment. The results will be documented in the procedures. Each test procedure will include the name, date/time occurred, and conditions encountered by the tester as the test cases are executed.
- **System Test Results Summary** – System test case results will be summarized in a System Test Results Summary report. The report will include a high-level summary of each test procedure objective, test case results, and test results. In the event of a failed or inconclusive test, the conditions under which the test failed will be described and the remediation action taken. Each remediation action shall be indexed and tracked as part of the acceptance process. The System Test Results Summary is a subset of the Operational Readiness Testing Plan results.
- **User Acceptance Testing Summary** – These are informal activities conducted by volunteers in an unstructured test environment. UAT participant profiles and comments will be collected, logged, and addressed as needed. The UAT testing environment enables input by users to document any issues. The UAT summary will be published periodically. High priority issues will be submitted as a ticket via the Jira backlog. The summary will be published periodically during the UAT period. It will continue during Phase 3.
- **Operational Readiness Testing Plan** – The Operational Readiness Test Plan (ORTP) provides a summary of the set of tests and supporting demonstrations that together establish operational readiness. The System Test Results Summary is a subset of the Operational Readiness Testing Plan results. It also includes the UAT summary results.

Most testing documentation will be stored in the project's Team folder. Agile sprint and release testing activities will be verified in Jira and transferred to test artifacts that are stored in the project Teams folder.



# 3 Agile Test Processes

This section describes the Agile testing processes including unit, integration, pre-production and production testing. A set of definitions that describes the Agile process including cycles, procedures and people are described in Table 4 and used throughout this section.

**Table 4. Definitions of Agile Methodology Terms.**

| Term               | Description   |
|--------------------|---|
| Scrum Team         | A scrum team includes a group of individuals, including a scrum master, product owner, and developers, collaborating to complete and deliver software products.   |
| Scrum Master       | A scrum master guides the scrum process and supports the scrum team, product owner, and organization.   |
| Product Owner      | Accountable for effective product backlog (requirements) management which includes that the product is accomplished and value is delivered.   |
| Sprint             | The sprint is the phase during which the scrum team completes development and testing on the user stories that have been placed in the sprint backlog.  |
| Grooming           | Obtain and define requirements for the development including acceptance criteria for verifying and validating code.   |
| Definition of Done | <p>The definition of done may be customized depending on specific project requirements, but should always include the following items:</p> <ol style="list-style-type: none"> <li>1. All acceptance criteria have been met.</li> <li>2. Code has been peer reviewed.</li> <li>3. Code has passed unit testing.</li> <li>4. Test plan &amp; test cases have been written and executed.</li> <li>5. Testing has been completed.</li> <li>6. All associated blockers and major issues have been resolved.</li> <li>7. The story has been reviewed and approved by the product owner (or proxy).</li> </ol> |

## 3.1 Agile Test Processes Overview

The Agile processes include story-based unit and integration testing. Testing is embedded in each product backlog item that is included in the 3-week sprint. The sprint and release cycles consist of the following processes.

1. Throughout a sprint, stakeholders, subject matter experts and the Product Owner (PO) contribute to defining the requirements that will be addressed through the development cycle. Inputs include:
  - To do list
  - Grooming (definition of each user story objective and acceptance criteria)
  - PO Review
  - Ready to sprint
2. During the sprint, stakeholders verify any potential bugs or raise issues and/or refinements to requirements. Key stakeholders join the demonstration sessions at the end of each sprint. (Multiple user stories may be demonstrated during a session). Sprint states include:
  - In progress
  - Code review
  - User story testing (unit) and integrated into code base (integration)
    - If the sprint is incomplete or a defect is detected, a ticket is generated that goes into the product backlog
  - User story verified (against acceptance criteria)
3. After sprint testing, stakeholders participate in release testing, and provide inputs/feedback. Testing tasks include:
  - Release demonstrations to solicit stakeholder feedback
    - If a defect is detected or requirement change is identified, a ticket is generated that goes into the product backlog
  - Release verified by demonstration
  - Release is “Done” and ready for burn-in
4. After each release, the software is field tested in a controlled, pre-production environment. Testing tasks include:
  - Formal pre-production testing (Alpha testing) of release
    - If a defect is reported, a ticket is generated that goes into the product backlog
  - Ready for production testing
  - Production (beta) testing
    - If a defect is detected, the Test Manager determines the source of defect. If software is related, then a ticket is generated that goes into the product backlog

In summary, each user story includes a set of acceptance criteria that validates the story requirement. Unit testing verifies the correct functionality and operations of the code. Integration testing verifies that the story is fully integrated with the existing code base and does not cause defects to the existing code base. Finally, after each sprint, the stories are demonstrated to confirm the acceptance criteria and verify correct functionality. If acceptance criteria cannot be met, the user story or element of the user story is reissued as a “ticket” back into the product backlog. The unit and integration verification are tested by automated scripts while the demonstration validates that the acceptance criteria are met.

Pre-production testing ensures that the release works with the hardware or simulated / third party testing tools (e.g., bus in a box, a tool that simulates bus dynamics). It includes formal, manual testing of the user facing systems (e.g., mobile apps, web app). Production testing ensures that the release works in a stakeholder environment. Production testing, also called User Acceptance Testing, combines all the elements of the system.

## 3.2 Unit and Integration Testing

The unit and integration testing are completed in order for any user story to be completed. The testing must run without a flaw and meet the story objective and acceptance criteria. The testing tools that are used to conduct unit and integration testing for the Buffalo ITS4US system include the following:

- **Unit testing** uses Jest scripts side-by-side with the corresponding code file in GitHub.
- **Integration testing** uses Appium tests at the root of the mobile/web code repositories providing end-to-end automated testing across the code base.
- **DevOps environment** uses GitHub to store and version control the source code. GitHub Actions allows test suites to run on each commit (operations that sends the latest changes of tested source code to the code repository).

**Unit testing** is performed using scripted and automated testing. Unit testing verifies the:

- Inputs and outputs of each component through API mocking which simulates actual APIs to respond with scripted responses for different scenarios – testing typical and unexpected behavior and dependencies.
  - Testing uses Mock Service Worker to generate mock APIs. The simulated triggers and actual responses are stored in the mobile and web code repositories on GitHub.
- Functional behavior for both applications and web services that correspond to the user story acceptance criteria.

**Integration testing** consists of end-to-end automated testing with scripts that simulate realistic user interaction with the actual screens. The service BrowserStack runs the Appium scripts on real devices using <https://www.browserstack.com>.

## 3.3 Issues Tracking and Testing

Issues due to bugs and defects are tracked as a new ticket in the product backlog, that is, as a user story. The issue priority is defined by a ticket type and scheduled based on its priority. The Product Owner sets priorities.

Every activity is issued as a ticket. Tickets range from user stories, tasks, to bugs. When issued, the tickets become part of the product backlog or work to be done to complete the Agile process. All tickets are assigned a priority ranking as described in **Table 5**. Most user story and task items

are designated a P4 or Normal Priority Ticket, for example, groomed and ungroomed user stories, research exercises and technical / performance reviews are assigned a P4 ticket. A defect, depending on its severity will be assigned a critical (P1), high priority (P2), medium to high priority (P3) ticket, or normal priority (P4).

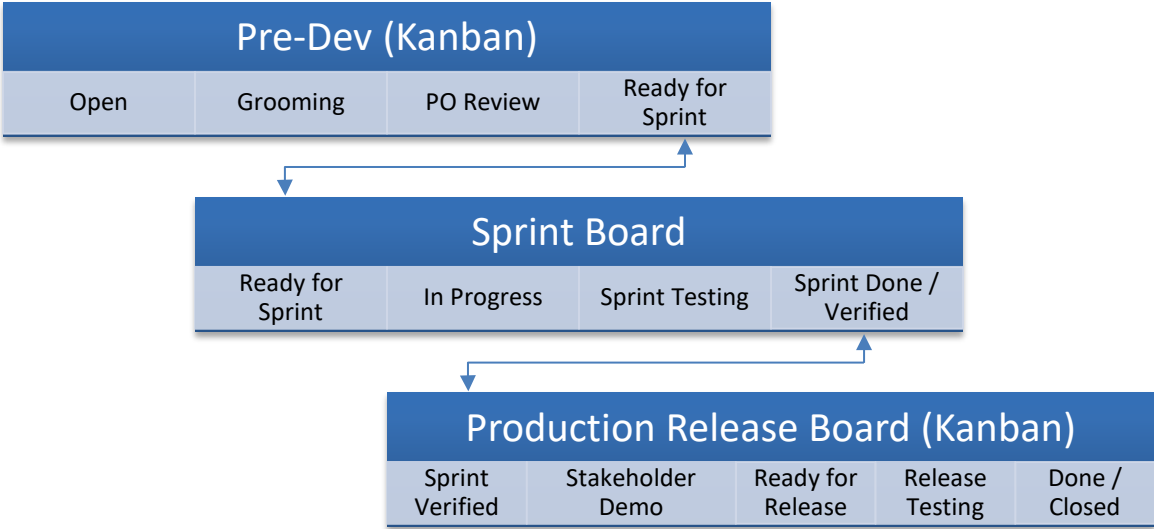
**Table 5. Ticket Types.**

| Ticket Type | Priority                       | Description  |
|-------------|--------------------------------|--|
| P1          | Critical Ticket                | System is not responding or system outage. Unscheduled downtime.   |
| P2          | High Priority Ticket           | Major feature or functionality within the system is not working. Causing a critical disruption for users. Must be fixed immediately.   |
| P3          | Medium to High Priority Ticket | Key defect or functionality that should be embedded into the <b>Active Sprint</b> to be worked on and completed before the sprint end date. These are the tickets that were not part of the sprint planning meeting but were identified after the sprint started and must be completed before the sprint end date. |
| P4          | Normal Priority Ticket         | Defect or functionality that should be reported to the scrum master and go through the grooming process. The scrum team will prioritize work items.  |

### 3.4 Sprint Reviews and Demos

The release cycle is composed of several Pre-Development and Sprint Board iterations as depicted in **Figure 13** (Note: this process is described in more detail in the System Design Document. Only the activities related to verification and validation are described in this document). When a sprint is ready and developed, a test engineer will generate one or more scripts to test the new code and merge it with the existing codebase (as described in Section 3.2). At each sprint the automated tests will exercise previous tests to ensure the merge is verified. After every sprint, the function is validated by demonstrating the function to the Product Owner and other key stakeholders to confirm compliance with the acceptance criteria and identify any functions that require additional research or were not accomplished.

When the software is ready for release, a larger demonstration with key stakeholder groups are convened to demonstrate the functions, verify the operations and validate the requirements and acceptance criteria. In addition, improvements and clarification are solicited and assigned tickets to add to the product backlog. These tickets are assigned priorities by the Product Owner.



**Figure 13. Agile Process.**  
*Source: Buffalo, NY ITS4US*



# 4 System Integration / Pre-production Testing

## 4.1 Integration Testing Overview

Integration Testing consists of testing releases or components (hardware) of individual systems. The approach for turnkey systems such as commercial off the shelf (COTS) or software as a service (SaaS) differ from Agile products. Pre-production testing will be performed based on the specific component. The following components are defined:

- SDS pre-production testing.
- Smart Infrastructure System Integration testing for hardware and SaaS components for:
  - TIH
  - Ped-X gateway
  - Indoor Navigation
- System integration for Agile releases and related hardware / SaaS components.

## 4.2 SDS Pre-production Testing

Until the SDS vendor is on-board, only the following testing processes are known. The tools and resources needed to implement these processes will depend on the selected vendor solution and associated tools. Demonstration and field testing follow similar testing cases. The areas include four domains:

- Tactical or maneuvering behavior
- Operational design domain (ODD)
- Object detection and collision avoidance, and
- Fail mode behavior.

The ODD differs between the controlled environments and field environments in which the SDS will operate.

Twelve (12) use cases are identified for testing the domains. The 12 scenarios will follow similar procedures in both the demonstration and field testing environments. These include:

- Left / Right Turn Test
- Right Turn Test
- Four-way stop with conflict
- Four-way stop without conflict

- Shuttle Stop
- Stationary pedestrian identification
- Moving pedestrian identification
- Following vehicle
- Leading vehicle
- Passing vehicle
- Object Detection
- Static Vehicle Obstruction

Specific data will be used for evaluating test use cases. Conduct of the demonstration (Section 4.2.2) versus field testing (Section 4.2.3) are described below.

### 4.2.1 Inspection Testing

Before the deployment of the SDS on the BNMC, the SDS vendor will simulate the operational environment or digital twin to inspect the functionality, operations, and performance of the SDS software and hardware. To apply the simulation, the vendor will generate data sets for testing. Specifically, the vendor will perform the following steps and validation checks:

**Step 1 - Map Creation and Data Collection:** For the BNMC SDS deployment area, the vendor is expected to generate the following datasets and check against the actual environment:

- Point Cloud Maps providing a detailed 3D point cloud of the route.
- Vector Maps which give a detailed vector representation of the route, including information about junctions, traffic rules, lanes, connections, traffic signals, etc.
- Bus stop locations and route definitions.

**Step 2 - Digital Twin Testing:** With the aforementioned datasets collected, the SDS fundamental use case and route are tested in the digital twin environment.

The map collection will support demonstration of the requirements identified in the SyRS included in the procurement.

### 4.2.2 Demonstration

Demonstration involves physical testing of the SDS in a controlled field test in the BNMC area, but without live passengers (i.e., with only the project team personnel onboard the shuttle). Demonstration testing will include:

- Step 1** - Testing and validating the maps created during Inspection Testing.
- Step 2** - Testing and validating the simulated use cases described by the test procedures.
- Step 3** - Collecting live route data for desktop hardware-in-the-loop testing.
- Step 4** - Calibrating path parameters and software parameters based on the fixed route, using the physical data collected in the previous step.
- Step 5** - Performing complete automated driving tests along the route, repeating calibration (Step - 4) if needed.

This activity may also involve testing in a safe, off-road environment (i.e., test tracks). These test tracks could be ones to which the vendor has access, or they could involve the University at Buffalo Proving Testing Grounds for Connected Automated Vehicles (CAVs).

### 4.2.3 Field Testing

The final stage, which follows the demonstration (simulation and test-track testing), will be field testing or testing the functionalities of the SDS, the SDS SOC, and the CTP integration with live passengers during the last three months of Phase 2. During this phase of testing, a number of safety protocols and precautions will be put in place including:

- The vehicles will always have trained safety stewards at the wheel with the ability to take control of the vehicle at any moment.
- Safety stewards will have the capability to abort testing if they feel that safety is compromised.
- Testing will initially be conducted during off-peak hours when the operations is least likely to disrupt traffic, and when there are fewer vehicles on the road.
- Any issues discovered during this phase will be documented and reported to create fixes and to confirm the ODD. Any modifications to the performance will be tested first in the controlled environment.

## 4.3 Smart Infrastructure System Integration / Pre-production Testing

The Smart Infrastructure (SI) will test three turnkey or SaaS systems. They include:

- Transportation Information Hubs (TIH)
- Pedestrian Crossing Request Actuation (Ped-X)
- Indoor Navigation

The three SI systems include hardware components which require bench testing. In addition, internal software functions are implemented as SaaS and testing includes verifying proper operations and validating operations against the requirements. The SDS and HDS reservations APIs include similar SaaS reservation-schedule-dispatch (RSD) functions that are tested using the same system integration testing processes.

During pre-production testing two major test activities occur:

- Inspection or bench testing applied to system components of the SI systems including hardware and system function and performance requirements.
- System integration testing between the CTP and SI or RSD functions. These integration tests allow the verification that the systems perform according to expectations in end-to-end testing.

### 4.3.1 Inspection / Bench Testing

Inspection or bench testing exercises the functionality and configuration of the hardware or software prior to installation.

Hardware environmental, reliability or performance testing requires vendors to provide certification of conformance testing from an accredited organization. Procurement of these systems requires proof of conformance.

The test procedures ensure that all the specified functionality, performance and operations work properly with a variety of scenarios described in test procedures. All testing will be documented and summarized in the **System Test Results Summary**. Specifically, inspection test procedures for the:

- **TIH** will exercise the functionality and performance of the hardware components, e.g., monitor, keyboard, communications, security components, and processor.
- **Ped-X** will exercise the functionality and performance of the hardware components – the traffic signal gateway. The SaaS components will be reviewed and configured for the specific intersections where the gateway will be deployed.
- **Indoor Navigation** will consist of a pilot conducted at two test facilities to calibrate the system by assessing sensor signal sensitivity and different handset detection capabilities. These tests will be replicated in field tests at the Visual Impairment Advancement (VIA) and Kaleida Health facilities. In addition, all hardware components (i.e., beacons) will be inspected for battery and operational efficacy prior to installation.

There are no hardware elements associated with the RSD SaaS tools.

### 4.3.2 System Integration Testing

System integration testing performs “black box” testing to verify *end-to-end* testing results based on expected outcomes for various scenarios. During the Agile process, each SI and RSD interface will be tested in a simulated environment using a range of scenarios that test boundary conditions, anomalies and functionality. During integration testing, operations and performance will be tested to ensure that the system produces the expected results across all systems. At each release alpha testing additional components will be tested along with a sample of previous test cases to verify expected outcomes. By the third CTP release all the components will be fully integrated, so that **comprehensive system integration testing** will be conducted during Alpha Testing for Release 3. Release 4 will comprise bug fixes and adjustments to the full system.

System test cases will be drafted into a formal system test procedure and will be manually conducted from five to ten times to test repeatability while applying different testing conditions or simulated conditions.

Each release will be tested on every channel that has been implemented within the release:

- Web – applies to all four Releases. Web browsers include: Edge, Safari, Chrome, and Firefox.
- Mobile– applies to all four Releases. Mobile platforms include iOS and Android.
- TIH – applies to Releases 3 and 4.
- Call Center – applies to Releases 3 and 4.

The test procedures generated for each release will depend on the functionality incorporated in the release. The functionality is driven by the sprint priorities. The Software Development Schedule includes the functionality expected at each release. After each release, system integration testing is performed with field equipment. A three week stage/pre-production testing follows the CTP / PMD release. The stage testing is performed using manual procedures and test cases under a variety of test conditions. These are typically referred to as Alpha tests.

**Table 6** shows the functionality associated with each CTP release and SI, CS, PMD or external system component that will be tested during the Alpha test procedures. Although each set of functions in the release will be field tested, not all functions include an integration element.

**Table 6. System Integration Testing.**

| Release | Release Testing Functions                     | Dependencies:<br>Application / Hardware<br>Integration Elements   |
|---------|---|---|
| -       | CTP Release MVP Alpha Testing (CTP MVP)       | -   |
| CTP MVP | User Management                               | -   |
| CTP MVP | Data Layer                                    | External data feeds (OSM, GTFS)   |
| CTP MVP | Trip Planning                                 | -   |
| CTP MVP | Website MVP                                   | -   |
| CTP MVP | Mobile App MVP                                | -   |
| CTP MVP | Data Feed and Enrichment                      | -   |
| -       | CTP Release 2 Alpha Testing (CTP R2)          | -   |
| CTP R2  | Trip Reservations                             | Integration with HSD / SDS reservation API  |
| CTP R2  | Notifications                                 | -   |
| CTP R2  | Special Data Feed Integration                 | Detour and traffic information (as needed)  |
| CTP R2  | Third-party Accounts                          | PMD Survey API integration  |
| CTP R2  | Trip Planning: Status                         | -   |
| CTP R2  | Account Removal and PII (includes Metric API) | Integration with PMD data ingestion   |
| CTP R2  | Indoor Navigation                             | Integration with Indoor navigation sensors and SDK  |
| CTP R2  | UI: Website, Mobile, Kiosk                    | Integration with TIH  |
| -       | PMD Release 1 Alpha Testing (PMD R1)          | -   |
| PMD R1  | PMD Environment                               | Connection between UB server and external data sources, including the UB server which stores the survey data. |
| PMD R1  | PMD Ingestion Processes                       | CTP Metrics API integration   |
| PMD R1  | PMD UI  | -   |
| -       | CTP Release 3 Alpha Testing (CTP R3)          | -   |

| Release | Release Testing Functions                                       | Dependencies:<br>Application / Hardware<br>Integration Elements |
|---------|---|---|
| CTP R3  | Service Animal  | -   |
| CTP R3  | Ped-X   | Integration with Ped-X at Ellicott and High                     |
| CTP R3  | Call Center   | Integration with Call Center terminals                          |
| CTP R3  | Language  | -   |
| CTP R3  | Group Travel  | -   |
| CTP R3  | Caregiver / Trusted Companion                                   | -   |
| CTP R3  | Accessibility   | -   |
| CTP R3  | UI: Website, Mobile, Kiosk                                      | Additional integration with TIH and call center                 |
| -       | PMD Release 2 Alpha Testing (PMD R2)                            | -   |
| PMD R2  | PMD Refinement of CTP Metrics                                   | Additional integration with new data from CTP Metric API        |
| PMD R2  | PMD Ingestion Processes   | SDS, HDS, NFTA data integration                                 |
| PMD R2  | PMD UI Visualizations   | -   |
| -       | CTP Release 4 Alpha Testing (CTP R4)                            | -   |
| CTP R4  | O&M   | -   |
| CTP R4  | CTP Security  | Penetration testing   |
| CTP R4  | CTP Reliability   | -   |
| CTP R4  | Self-monitoring Log   | Review monitoring logs  |
| CTP R4  | Application and Feature Expansion                               | As needed   |
| -       | PMD Release 3 Alpha Testing (PMD R3)                            | -   |
| PMD R3  | PMD Refinement of CTP Metrics                                   | Additional integration with new data from CTP Metric API        |
| PMD R3  | Collect and Refine Sample Datasets from UAT (Beta participants) | -   |
| PMD R3  | O&M   | -   |

## 4.4 Full System Testing

Full system integration testing relies on design coordination, integration, installation, and testing activities. All the system elements will be fully integrated at the conclusion of the CTP Release 3, PMD Release 2, pre-production testing for SDS, Ped-X, HDS RSD, INS, and TIH. At this point, the full, comprehensive testing will commence. The test case types described in Section 4 will be updated and conducted to validate the requirements and verify operations. In addition, several additional use cases will be conducted to validate safety requirements and address degraded modes of operations including but not limited to:

- Degraded with communications or data feeds from external systems, smart infrastructure or CTP.
- Degraded with communications or data feeds from smart infrastructure or CTP to Community Shuttle (particularly the SDS).
- Degraded when community shuttle services dependent on other subsystems are in reduced availability or impacted.

The results of these test cases will be assembled and documented in the System Test Case Results and System Test Results Summary.

## 4.5 System Dependencies

There are several factors that add to the success of these activities. The following practices will drive the coordination activities listed in **Table 6**.

- Publication of interfaces which include its characteristics and behavior (e.g., structure and semantics) that is available to each integration partner to access interface changes as the interface description is refined.
- Development / testing environment (also called “sandbox” or pre-production testing environment) that includes the ability to generate interface scenarios (content and conditions) that simulate real-world factors.
- Installation of most hardware elements prior to pre-production testing.
- Developing a realistic schedule throughout the developing and testing phases, releases and other milestones.
- Coordination meetings through regular Agile or testing ceremonies to address risks and “blockers” to achieving the milestones.

The dependencies listed in Table 6 (Dependencies column) provide a high-level set of dependencies associated with integrating the subsystems and applications. The Agile process is used to identify integration elements that need to be in place prior to incorporating into the software development (Agile releases) and during integration testing. In addition, installation of equipment (e.g., SI, SDS field testing) also needs to be in place prior to most pre-production testing.



# 5 User Acceptance Testing

User Acceptance Testing (UAT) is conducted to accept the system. It is one of the final tests administrated by users to detect defects and identify anomalies in the system.

## 5.1 UAT Testing Overview

User Acceptance Testing is conducted following successful major releases and pre-production testing of all the system elements. The UAT will consist of user beta testing by the initial set of recruited participants. Beta testing will consist of the following and be consistent with the provisions of the Phase 2 Performance Measurement and Evaluation Support Plan (PMESP) and Human Use Agreement (HUA).

The beta testing will be conducted with trained participants, about 100 people from the various stakeholder groups. Informal taining on the systems will proceed prior to testing (April) and throughout the recruitment period. Participants will be provided with a beta website and mobile app to use during the test. On-line surveys and bug software (e.g., Bugsee) will be installed on the app for feedback and bug reporting. In addition, follow up outreach efforts will be put in place to solicit additional information on anomalies, defects and improvements for the mobility services.

To that end, the UAT will follow pre-production testing of

- CTP Release 3.
- PMD Release 2.
- Installation and pre-production testing of TIH, Ped-X, indoor navigation elements.
- Integration of HSD/SDS reservations, scheduling and dispatch services with the CTP.
- Pre-production testing of SDS without passengers.

Full system testing (end-to-end) testing of all subsystems, applications and components, as well as operations and maintenance procedures for the Buffalo All Access system. User Acceptance Testing consists of the following activities:

- Pre-production phase testing (all components).
- Production testing platform is set up and ready; environment includes functions to ingestion feedback from Beta testing participants (including defects).
- Participants have been recruited and trained to operate systems (safely) and report defects.
- Schedule and facilitate UAT feedback sessions to review experience.

- Mechanism is set up to update software / hardware elements (especially those due to critical or high-level defects).

Operational and maintenance procedures are established and activated during UAT period which includes staffing of community shuttle, call center and other facilities needed to support customers. In addition, the Operational Readiness Plan (ORP) and Demonstration will be completed during the UAT phase. The details of the plan and demonstration are described in the ORP.

The beta period will commence in late April 2024 and extend through the end of Phase 2.

## 5.2 Reporting Defects

Defects will be reported throughout testing and tickets for software and software integration issues will be generated, or reported to vendors for turnkey systems or SaaS.

For UAT, CPT defects will be collected through the pre-production and production testing environments. Mobile app testing environments (for iOS and Android) include BugSee, an internal platform to record anomalies and generate reports. In addition, the web production test environment used also captures user detected issues for review. Other test environments for the community shuttle and smart infrastructure also contain forms that users may record their actions to recreate the defect and submit the impact of the defect. Each component's system developer will track the input.

Software as a service and turnkey systems including INS, Miovision, SDS, and TIH have similar environments that will capture defects and user issues to be documented and resolved.

# 6 Defect / Deficiency Tracking and Resolution Processes

This section describes how defects and user input will be tracked and resolved through the project.

## 6.1 Defect / Deficiency Categorization and Impacts on Testing

The defect and deficiency categorization will mirror the Agile process ticket types (see **Table 7**). The tickets will be expanded to incorporate priorities related to safety and hardware elements. The system level tickets correspond to all defects and deficiencies that occur in the Pre-production or production testing phases, as well as those identified during production and operations. The ticket type descriptions apply to all subsystems and components of the system. The system defects may generate Agile tickets when an Agile software component is identified as the issue. System defects will be tracked and reported in an independent Defect Tracking Log file (see Section 0 for more details).

**Table 7. System Level Defect/Deficiency Categories.**

| Ticket Type | Priority                | Description   |
|-------------|-------------------------|---|
| S1          | Critical                | System is not responding; system outage or system malfunction poses safety risk.<br><br><b>Unscheduled downtime.</b>  |
| S2          | High Priority           | Major component, feature or functionality within the system is not working. Causing a critical disruption for users.<br><br><b>Must be fixed immediately.</b>   |
| S3          | Medium to High Priority | Key defect or functionality that should be working in the short term (next sprint or release). These are tickets that are identified during or before pre-production testing.<br><br><b>Will be incorporated into an interim release or version prior to, or during production testing.</b> |
| S4          | Normal Priority         | Defect or deficiency in functionality that should be reported to appropriate developer, implementor or integration team to resolve.<br><br><b>Work items will be prioritized by respective development team.</b>  |

## 6.2 Defect / Deficiency Resolution

Defects and deficiencies will be identified and reviewed differently based on the component and testing phase. **Table 8** presents the testing phases and resolution process.

**Table 8. Testing Phases and Resolution Processes.**

| Subsystem / Testing Phase      | Testing Phase   | Resolution Process   |
|--------------------------------|---|--|
| PMD or CTP                     | Agile (Section 3)   | <p>A “P” ticket is issued during:</p> <ul style="list-style-type: none"> <li>• the sprint if a user story does not meet its acceptance criteria.</li> <li>• a sprint or stakeholder demonstration if stakeholders recommend improvements during a release or sprint.</li> </ul> <p>The prioritization of the ticket (see Table 5) is made by the Product Owner.</p>  |
| PMD, CTP, Integration Elements | Pre-Production Integration (Section 4.3.2)                | <p>The cause of the defect or deficiency is identified, replicated and reviewed to determine the fault or deficiency. These are software or interface components. Once the fault and impact are determined, the resolution of the issues are submitted to the respective component developer and resolved through their appropriate processes. For example, a PMD defect will follow their Agile testing processes until resolved.</p>   |
| SDS                            | Pre-Production Inspection and Demonstration (Section 4.2) | <p>The cause of the defect will be identified, prioritized and directed to the system vendor to resolve.</p>   |
| Other Hardware Systems         | Pre-Production Inspection (Section 4.3.1)                 | <p>The cause of the defect will be identified and directed to the system vendor to resolve.</p>  |
| System                         | UAT (Section 5)   | <p>The cause of the defect or deficiency is identified, replicated and reviewed to determine the fault or deficiency. Once the fault and impact are determined, the resolution of the issues are submitted to the respective component developer and resolved through their appropriate processes.</p> <p>Before the item is re-released, it will be subjected to integration testing. The severity of the issue and its role within the system may require different degrees of regression testing. These will be determined on a case by case basis.</p> |

## 6.3 Defect / Deficiency Tracking and Reporting

A process will be established to track and report on the status of defects and changes to the system. The process is illustrated in **Figure 14**. The tracking begins when a defect is reported during stage testing (during the pre-production) and recorded in the Defect Log, a spreadsheet that will be published on the Teams site.

### 6.3.1 Defect Review Board

A Defect Review Board (DRB) will be organized to manage and validate the Defect process. The following characterizes the composition and responsibilities of the board.

- The DRB is composed of representatives of each the developer teams.
- The DRB will be led by the Agile Product Owner (a representative from NFTA) or designated agent.
- A DRB will meet periodically based on the frequency of testing or when a critical or high priority failure is detected. For example, during Release testing, the team will meet at least once a week to review test reports and defects.
- Meetings of the DRB will be documented and published in Teams.
- A **Defect Tracking Log** will be managed by the DRB. The Test Manager will be assigned to enforce configuration control over the Defect Tracking Log.

### 6.3.2 Defect Process and States

The defect process will follow the activities depicted in **Figure 14**. As each process is concluded, a defect state is designated. Defect states include detected, opened, assigned, in-progress, closed, cancelled and reopened.

- **Detected:** prior to review by the Defect Review Board, when a test fails, or a defect is reported, it is logged into the defect log.
- **Open:** when reviewed by the Defect Review Board, a ticket is opened to address the defect. The team assigns a priority ranking to the defect. The review may include a solution to the failure.
- **Assigned:** after the defect is opened, the defect is assigned to a developer and / or tester. The technical team lead or scrum master typically assign the developer.
- **In-Progress:** when the developer begins working on the defect, the state is changed to in-progress. The state also includes completion and passage of all related test cases. Regression testing may be needed to approve the testing.
- **Closed:** when the testing results are reviewed and confirmed by the Test Lead and Test Review Team, they are presented to the DRB who closes the defect.

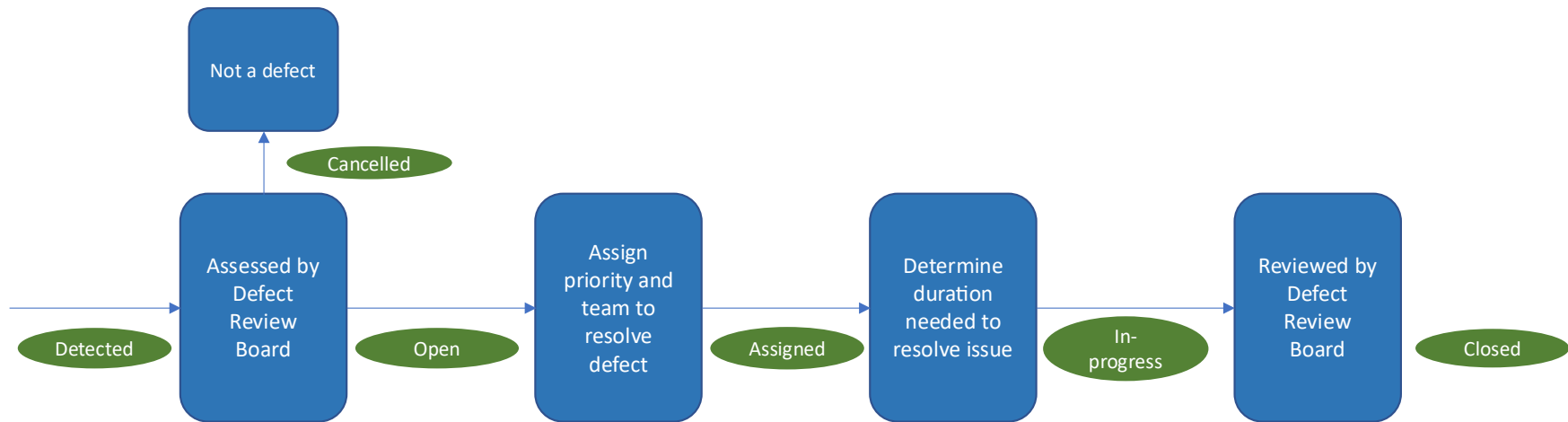
- **Cancelled:** due to refactoring or other change in system requirements or needs, the defect is obsolete, no longer occurs or was misreported. The DRB must agree that the defect is abandoned. A defect will be cancelled if one of the following assessment results is true:
  - The defect cannot be reproduced and there is no impact to any of the systems.
  - The defect is duplicated by another defect. Additional details of this report should be transferred to the other defect report.
  - The defect is determined to be a user error. Information on this report will be forwarded to training staff.
- **Reopened:** if a defect occurs again once closed, the DRB may reopen the defect item.

### 6.3.3 Defect Tracking Sheet

The Defect Tracking Log spreadsheet used will include the following information:

- **Index:** unique number used to track the defect.
- **Date Detected:** date when defect was detected.
- **Description:** summary of the defect that describes the impact of the defect.
- **Test type:** the type of test conducted that detected the defect (Alpha, Beta, operations (Phase 3)).
- **Component:** the component where defect was detected (i.e., CTP android, CTP iOS, CTP web, CTP call center, CTP kiosk, INS, Ped-X, PMD, etc.).
- **Comment (beta test form):** URL to a form reported by the users.
- **Test Procedure / Case / Line (alpha test procedure):** URL for the test report summary.
- **Categorization:** System level priority category (see **Table 7**).
- **Impact:** impact to the system (open content).
- **Assigned to:** person or team to which this defect was assigned.
- **Date Assigned:** date defect was assigned.
- **Status:** detected, opened, assigned, in-progress, closed, cancelled and reopened.
- **Date Closed:** Date when defect was closed.

The Defect Tracking Log will be updated on a daily basis and reviewed prior to each DRB meeting.



**Figure 14. Defect Tracking Process.**

*Source: Buffalo ITS4US Deployment Project*

### 6.3.4 Defect Reporting

A defect report will be automatically generated when the spreadsheet is updated. Each defect report will contain metrics derived from the Defect Tracking Log. The defect report will contain the number of defects by:

- State
- Component
- Test Type
- Category
- State and Component by duration that is not closed or cancelled

Additional information may be added to the test report after review by the DRB.

# 7 Requirements and Test Case Traceability List

Test cases are tracked to system to validate that all the requirements have been implemented and operate as expected. The test cases and summary results are also inventoried and tracked in a matrix to verify proper operations of the system. The validation and verification matrix that describes the test and tracks its success are included in the table below. The table fields are described as follows:

Editor's Note: as an Agile development, the test cases are updated following each release. The test cases and results are currently stored in ITS4US - Buffalo - Master Test Tracking.xlsx and will be inserted into this document for the final deliverable.

- **Req ID:** requirement identifier
- **Test ID:** test case identifier
- **Test Phase:** The test phases where test is conducted. These include Agile (unit, integration), Pre-production (inspection/bench, system integration) and User Acceptance testing
- **Test Case:** A list of test cases that verify requirement(s).
- **Platforms:** platform on which pre-production testing (alpha) of subsystems and applications are conducted. These include:
  - CTP web includes test procedures conducted on Web browsers (Chrome, Safari, Edge, Firefox)
  - CTP mobile includes test procedures conducted on Mobile apps (iOS, Android, iPad)
  - INS
  - Ped-X
  - SDS
  - HDS RSD
  - TIH
- **Verification Method:** verification method used include methods described in Section 2.4.
- **Related req'ts:** requirements that are related to the test.



# Appendix A. References

- [1] [ConOps] Gopalakrishna, D., et al. (2021). *Phase 1 Concept of Operations (ConOps) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-860)*. Federal Highway Administration.
- [2] [DMP1] Gopalakrishna, D., et al. (2021). *Phase 1 Data Management Plan (DMP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-868)*. Federal Highway Administration.
- [3] [DMP2] Gopalakrishna, D., et al. (2023). *Phase 2 Data Management Plan (DMP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-868)*. Federal Highway Administration.
- [4] [DPP] Gopalakrishna, D., et al. (2023). *Phase 2 Data Privacy Plan (DPP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-868)*. Federal Highway Administration.
- [5] [SMP] Gopalakrishna, D., et al. (2021). *Phase 1 Safety Management Plan (SMP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-873)*. Federal Highway Administration.
- [6] [PMESP] Gopalakrishna, D., et al. (2021). *Phase 1 Performance Measurement and Evaluation Support Plan (PMESP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-878)*. Federal Highway Administration.
- [7] [SyRS] Gopalakrishna, D., et al. (2021). *Phase 1 System Requirements Specification (SyRS) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-883)*. Federal Highway Administration.
- [8] [SEMP] Gopalakrishna, D., et al. (2021). *Phase 1 System Engineering Management Plan (SEMP) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-21-918)* Federal Highway Administration.
- [9] [SAD] Okunieff, P., et al. (2023). *Phase 2 System Architecture Document (SAD) – Buffalo, NY ITS4US Deployment Project (FHWA-JPO-22-981)* Federal Highway Administration.
- [10] [SDD], TBD



# Appendix B. Acronyms

**Table 9. Acronyms**

| Acronym | Description   |
|---------|---|
| API     | Application Programming Interface                       |
| BNMC    | Buffalo Niagara Medical Campus                          |
| CAV     | Connected Automated Vehicles                            |
| CoB     | City of Buffalo   |
| ConOps  | Concept of Operations                                   |
| COTS    | Commercial off-the-Shelf                                |
| CPT     | Continuous Performance Testing                          |
| CS      | Community Shuttle                                       |
| CTP     | Complete Trip Platform                                  |
| DMP     | Data Management Plan                                    |
| DOT     | Department of Transportation                            |
| DPP     | Data Privacy Plan                                       |
| DRB     | Defect Review Board                                     |
| ET      | Eastern Time  |
| FHWA    | Federal Highway Administration                          |
| FTA     | Federal Transit Administration                          |
| GBNRTC  | Greater Buffalo-Niagara Regional Transportation Council |

| Acronym | Description  |
|---------|--|
| GTFS    | General Transit Feed Specification                             |
| HDS     | Human-Driven Shuttle   |
| HUAS    | Human Use Approval Summary                                     |
| ICD     | Interface Control Document                                     |
| ID      | Identification   |
| IE      | Independent Evaluator  |
| iOS     | iPhone Operating System  |
| ITS JPO | Intelligent Transportation Systems Joint Program Office        |
| ITS4US  | Intelligent Transportation Systems for Underserved Communities |
| MSA     | Metropolitan Statistical Area                                  |
| MTBF    | Mean Time Between Failure                                      |
| MVP     | Minimum Viable Product   |
| NFTA    | Niagara Frontier Transportation Authority                      |
| NITTEC  | Niagara International Transportation Technology Coalition      |
| ODD     | Operational Design Domain                                      |
| ORP     | Operational Readiness Plan                                     |
| ORTP    | Operational Readiness Test Plan                                |
| OSM     | Open Street Map  |
| OST     | Office of the Secretary  |
| OTS     | Off-the-Shelf  |
| PAL     | Paratransit Access Line  |

| Acronym | Description                                     |
|---------|---|
| Ped-X   | Pedestrian Signal Request                       |
| PII     | Personally Identifiable Information             |
| PM      | Performance Measure                             |
| PMD     | Performance Management Dashboard                |
| PMESP   | Performance Measurement Evaluation Support Plan |
| PO      | Product Owner                                   |
| PROW    | Public Right of Way                             |
| RSD     | Reservation-Schedule-Dispatch                   |
| RSG     | Resource Systems Group                          |
| RTM     | Requirements Traceability Matrix                |
| SaaS    | Software as a Service                           |
| SDK     | Software Development Kit                        |
| SDS     | Self-Driving Shuttle                            |
| SI      | Smart Infrastructure                            |
| SOC     | Shuttle Operations Center                       |
| SOI     | System of Interest                              |
| STP     | System Test Plan                                |
| SyRS    | System Requirements Specification               |
| TBD     | To Be Determined                                |
| TIH     | Transportation Information Hub                  |
| UAT     | User Acceptance Testing                         |

| Acronym | Description                                |
|---------|--|
| UB      | University at Buffalo                      |
| UI      | User Interface                             |
| URL     | Uniform Resource Locator                   |
| USDOT   | United States Department of Transportation |



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](http://www.its.dot.gov)

FHWA-JPO-23-995



U.S. Department of Transportation