n-catt
National Center for Applied Transit Technology

A Guidebook on New Software Adoption for Small Transit Agencies

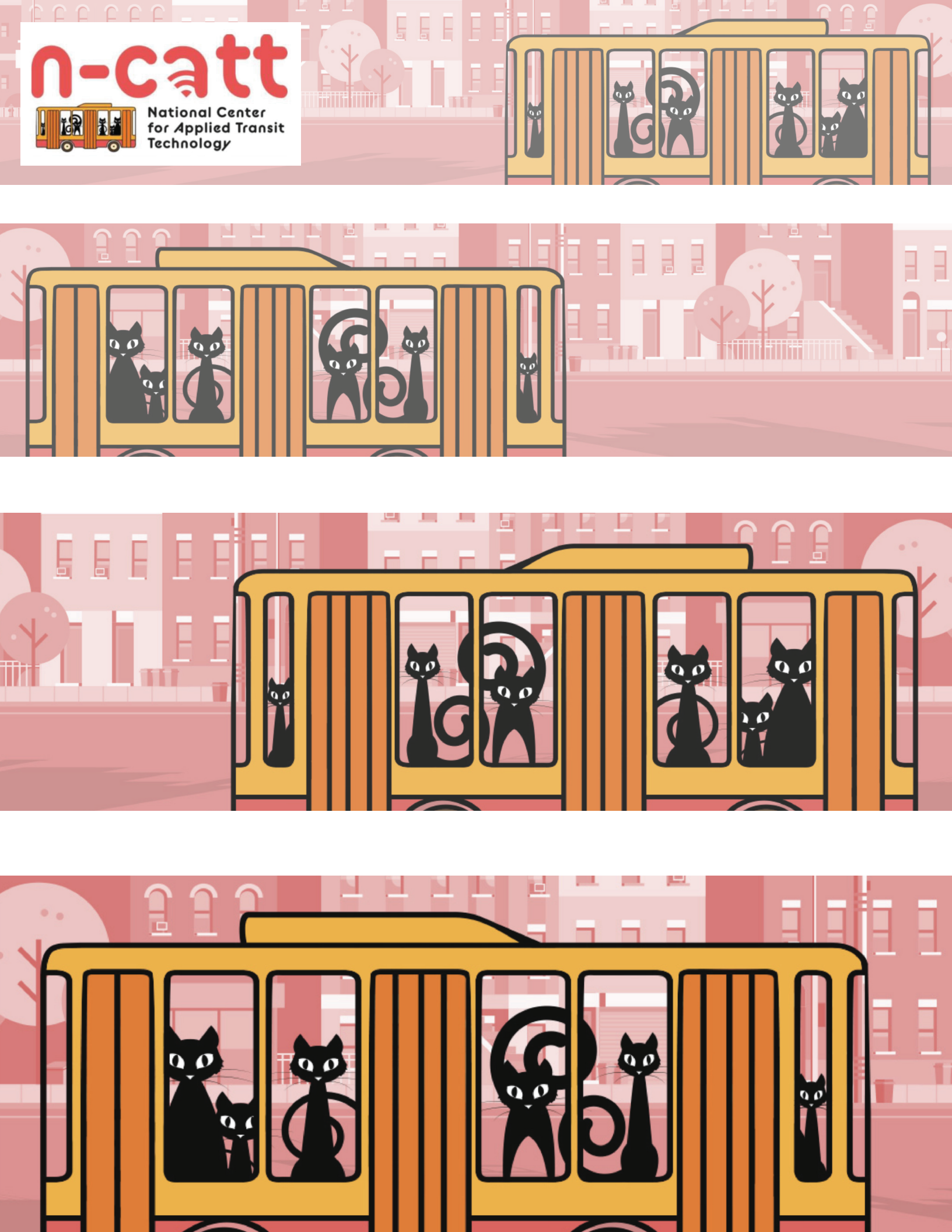
March 2021

Image sources: Mountain Line Transit Authority located in Morgantown, West Virginia (top left and right), North Central Regional Transit District located in Española, New Mexico (bottom left), and EZ Ride located in New Jersey (bottom right)

Prepared by

DEMANDTRANS®
SMART • MOBILITY • SOLUTIONS

Civic Sphere


n-catt
National Center
for Applied Transit
Technology

Contents

Executive Summary 1
Introduction and Background Information 6
Guidebook Purpose and Structure 6
Guidebook Focus Areas and Software Types 7
Software Types and Related Concepts 8
Software for Demand-responsive Transit Operations 8
Software for Multiple Agencies 10
Software for Integrated DRT Service 10
Guidebook "Quick Content" and Worksheets 11
About N-CATT 11
About this Document 12
Chapter 1 / Step 1: Set the Software Scope 13
Step 1a. Clarify the Software's Purpose 13
Step 1b. Identify General Software Connectivity Needs 14
Step 1c. Anticipate Resources to Apply to Software Adoption 16
Key Takeaways 17
Illustrative Project 17
Chapter 2 / Step 2: Collaborate with the Software Stakeholders 20
Step 2a. Create a Stakeholder Map 21
Step 2b. Identify Key Topics for Each Stakeholder Group 23
Step 2c. Create a Tailored Information-gathering Process to Integrate Stakeholder Findings 23
Key Takeaways 25
Illustrative Project 26
Chapter 3 / Step 3: Move Forward with a Software Product 28
Available Software Product Types 28
Computing Platform Considerations 31
Software Product Purchasing Options 32
Software Functional Types for Small Transit Systems 34
Software Requirements, Functions, and Features 39
Software "Ownership" Requirements 39
Software Affordability 41
Software Application Procurement 43
Step 3a. Determine What Type of Software Your Agency Needs 44
Step 3b. Understand Your Available Software Choices 44

Contents

Step 3c. Determine Whether to Obtain a SaaS System or a Licensed Software Product 44
Step 3d. Determine Your Core Requirements for the Software 45
Step 3e. Develop the Request for Proposals 45
Step 3f. Evaluate the Proposals and Select the Most Appropriate Software Product 46
Step 3g. Begin the Software Implementation Process 47
Key Takeaways 47
Chapter 4 / Step 4: Support the Software 48
General Software Support Needs 48
Step 4a. Plan for One-Time Software Setup and Training 51
Step 4b. Prepare for Ongoing Support Needs 52
Step 4c. Consider Additional Support as the Software Scope Expands 55
Adopting a Partnering and Learning Approach 56
Key Takeaways 57
Illustrative Project 58
Worksheets 60

Executive Summary

Guidebook Purpose

During the past decade, there has been a veritable explosion of software options that are available to small city/rural/tribal transit agencies to assist them in improving their operations and passenger interactions. These software applications span the full range of small transit agency services—encompassing fixed route bus services and traditional and new generation demand responsive transportation—and include customer-facing applications such as map-based views of bus location and schedule adherence, trip planning, self-service booking of demand responsive trips, and fare payment and ticketing as well as operationally-focused software used by the agency to manage its core services. Moreover, a number of new technology companies have entered the transit market with new generation software products, and in response established software providers are raising their games and expanding the capabilities of their products. There has never been a better time for a small transit agency to take advantage of software to help achieve its objectives and improve service to its customers.

There are two dimensions to the opportunities afforded by these developments. First, software can be applied to existing services and their operations as a means of increasing operational efficiency, bolstering productivity, and improving the passenger experience. Second, the capabilities of the newer generation of software may stimulate an agency to re-imagine how it serves its passengers, since some software increases the viability of modes and digital services that may be entirely new to an agency. Software has the potential to fundamentally transform transit agency services.

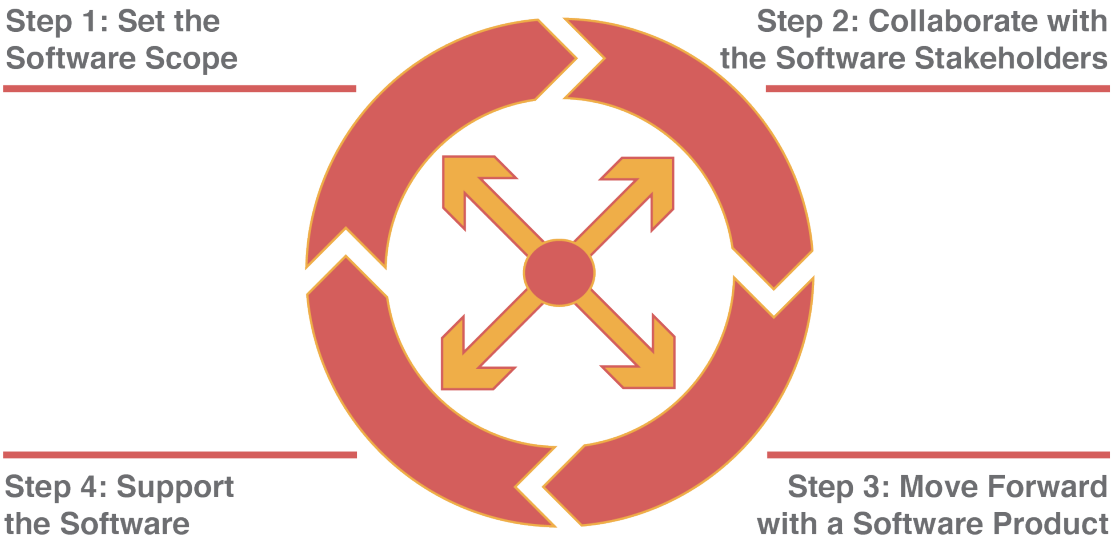
With these opportunities also come significant challenges to small transit agencies that wish to take advantage of new software. Whether the software application is focused on internal operations or direct customer interaction, it may lead to—and/or require—significant adjustment in agency processes. Not only will staff need to be educated on how to use the software, but the agency may also need to make important internal changes to accommodate entirely new day-to-day work tasks—even as other tasks are largely automated by the software. Moreover, as the capabilities of available software expand the possibilities of the agency, it may need to create new types of customer or organizational relationships to actualize these possibilities.

The purpose of this Guidebook is to enable you and your agency to navigate the way forward in an informed fashion vis-à-vis the enhanced use of software for your services and by your customers.

Guidebook Structure

New software adoption has the potential to range from a relatively simple undertaking to an extremely complex one. While the details of an agency's individual software adoption process are unique, specific to the intricacies of each agency applying it, the overall structure of the process can be generalized and broken down into a set of steps to follow. This Guidebook provides a four-step process to move from the initial stages of software consideration to later steps involving set-up, operations, and maintenance. The four steps are shown on the following page as a cycle of steps that often occur chronologically in the order shown, but do not necessarily follow such a sequence in all cases. The details of each step are contained within separate chapters.

The Guidebook closes with a series of worksheets. The worksheets walk a reader through the activities mentioned in each chapter, with clear guidance on how to apply the information. For some agencies, completing the worksheets individually or during a collaborative session could be the next step in better understanding how their own software adoption process could work. The worksheets help Guidebook users organize their thoughts, pinpoint gap areas, and plan for upcoming efforts.



Guidebook Focus and Software Types

The Guidebook primarily focuses on specific types of software—shown below—that are common for many small transit agencies. Of approximately 1750 small transit systems of less than 20 vehicles in the USA, over 1125 (65%) are providing some form of demand responsive transportation service and therefore require trip booking and scheduling capabilities (although for very small systems this function can often be accomplished without an actual software product designed specifically for this purpose). However, the Guidebook is applicable to essentially all software relevant to small transit agencies.

1. Trip planning – Trip planning software enables a customer to view their transportation options online, typically across most or all available choices such as fixed-route transit, demand-responsive transit, bike sharing, walking, and other modes that may be available in an area. This software type enables a customer to quickly compare their options across multiple factors such as trip duration, departure/arrival time, cost, and other factors. More advanced versions of this type of software support "trip chaining" or combining multiple legs of different modes in a single trip (e.g., bike to fixed-route transit or demand-responsive transit to fixed-route transit). This type of software is typically accessed through a website or mobile app.

2. Trip booking and scheduling – Trip booking software enables a customer to book a ride on a demand responsive transit (DRT) service and to be provided with information on when the vehicle will pick them up. The trip booking process may be done directly by the customer via a web-based or smartphone app-based application, or may be accomplished by an agent on behalf of the customer using an on-line booking application. Trip booking is often directly connected to trip scheduling in the software application.

Vehicle scheduling is a fundamental element of all transportation services, and takes different forms for fixed-route transit and DRT, with corresponding differences in the types of software required. For small fixed-route transit operations, the vehicle trips to be scheduled are those on a fixed set of routes with pre-determined timetables for the vehicles. Such schedules can often be managed in spreadsheets or very basic software that also includes driver scheduling.

DRT scheduling software manages the capacity of the vehicle fleet used in the service. It determines the routing of each vehicle and how customer pick-ups and drop-offs are sequenced; the scheduling system is at the heart of the software that manages a DRT system and there are many levels of sophistication in different software applications.

3. Trip payment – Trip payment software, accessed via a mobile app or a website, enables a customer to purchase "tickets" for transit services via digital mechanisms. Such software utilizes payment methods such as credit and debit cards, and may also have options for "unbanked" customers. In some cases, trip payment software is directly connected to trip planning and/or trip booking software (i.e., customer reviews options via a trip planner, books the preferred trip, and then pays in advance).

Chapter 1 / Step 1: Set the Software Scope

This chapter breaks down the process of setting the software scope into three activities.

1. Clarify the Software's Purpose – Clarifying the purpose of the software involves identifying which software type is needed. For service operation needs, it is usually clear to the agency's staff the type of software which is needed, but there may need to be a more formalized process if the functional needs span different types of software or if customer facing software is involved. A suggested approach for handling these types of situations is presented.

2. Identify General Software Connectivity Needs – Whether an agency is acquiring multiple types of software to meet its needs or is adding a new type of software to its current software ecosystem, it is essential that it determine how much connectivity between the software applications will be needed. If the different software applications need to exchange data with one another, then the software must be "interoperable". When interoperability is necessary, the agency must ensure that its software acquisition process makes this a high priority in terms of the scope of the acceptable software solutions.

3. Anticipate Resources to Apply to Software Adoption – Resources relevant to the software adoption process include (at a minimum) financial resources, staff resources, existing software and computing infrastructure assets, and collaborator resources—financial, staff resources, or assets from partner organizations.

An agency should create an inventory of its likely available resources early on in order to be prepared for later steps in the process. Substantial financial and staff resources will be necessary for the procurement process that is part of Step 3: Move Forward with a Software Product and may also be needed in Step 4: Support the Software for tasks such as training staff members on the software and maintaining it.

Chapter 2 / Step 2: Collaborate with the Software Stakeholders

This chapter discusses the importance of working effectively with stakeholders and how that can be accomplished. Ways to identify the stakeholders to include in the process, and to then to actively involve them, are provided. Identifying the full range of stakeholders is important both to ensure that the software being acquired has the necessary functionality—an outcome facilitated by all of the key software users being involved in the planning and acquisition process—and to establish the collaborative process needed for the adoption of the software to move forward smoothly and effectively. The software stakeholders can generally be

disaggregated into 3 categories, as explained below.

1. Managers and procurers – The management of a software adoption process involves guiding its direction and ensuring it remains on track. The software adoption lead, similar to a project management role, is responsible for ensuring that all the parts—the results from Step 1 through Step 4—connect. Procurement, which occurs under the direction of the software adoption lead, involves selecting the software in accordance with the agency's requirements, policies, and procedures, and typically involves both dedicated procurement staff and subject matter experts. The former ensures that the policies and procedures are adhered to, while the latter helps draft the detailed content in the request for proposals. Procurers and managers function on behalf of the user groups.

2. Users – The software user groups encompass everyone who will interact with the software; they should be identified up-front to ensure later steps of the adoption process take the viewpoints of all users into account. For some types of software, the user groups include both members of the public and agency staff. Each user group will have a different set of requirements based on their needs.

3. Influencers – Influencers are stakeholders that won't directly use the software, nor will they be procurers or managers, but they will provide input into the process. For example, it is common in human services transportation to include the advice of groups that represent older adults, individuals with disabilities, and others with specific mobility considerations.

The manager of the overall process, the software adoption lead, is responsible for creating a tailored information gathering approach to integrate stakeholder findings. These findings will be incorporated into the development of the procurement documents—integrating the key concerns of each stakeholder group into the process.

Chapter 3 / Step 3: Move Forward with a Software Product

Small transit agencies today have a choice among multiple types of software products that address the same functional needs of the agency. These include commercial off the shelf (COTS) products, open source/public domain software, and custom software solutions. The advantages and disadvantages of each is discussed in this chapter.

During the past decade, there has been a strong trend in business software in the direction of Software as a Service (SaaS) approaches. Among multiple advantages of a SaaS purchase, a major advantage for a small transit agency is that it does not have to concern itself with the computing infrastructure on which the software is hosted. This chapter discusses the relative merits of SaaS approaches compared to licensed software products that are hosted by the agency itself or for which the agency is directly responsible. For small transit agencies, software solutions that avoid the agency needing to be responsible for their own computing infrastructure are typically advantageous, assuming that broadband data access is available.

This chapter provides high level descriptions of the various functional types of software products that are relevant to small transit agencies and the roles they may play. This includes products for both fixed route and demand responsive services, and for operational use as well as those used primarily by the agency's customers. There is also a discussion on software inter-operability, which becomes increasingly important as agencies acquire multiple software products and/or wish to engage in programs—such as mobility management—with other organizations.

It is important for small transit agencies to understand just what it is that software ownership requires of them if that software is to be most effective in serving their purposes, and the chapter includes a relatively detailed explanation of this. This includes such topics as software implementation, configuration, staff training, and upgrades.

The affordability of the software product is obviously of major concern to a small transit agency, and the different costs associated with the acquisition and on-going use of new software are identified and explained. There is a comparison of the different costs for the purchase of a SaaS product and a comparable licensed software product that is hosted and managed by the agency itself.

The chapter concludes with a series of 7 steps that need to be navigated from the time when the agency decides it wishes to acquire new software until the point when the software becomes operational for the agency, enabling them to move forward with a software product.

Chapter 4 / Step 4: Support the Software

The acquisition of a software product is merely the beginning of the process of incorporating it into the operations and daily use by the staff and customers of a small transit agency. For the capabilities of the software to be of maximum value to the agency, the agency needs an effective approach to support the software.

Agencies are likely to obtain the greatest value from their software if they forge a strong partnership with their software provider, a process that should begin as soon as the decision has been made about which company's product will be purchased. Using the partnership approach as the underlying assumption about an agency's working relationship with the software producer, support for the software system will mostly involve the following 3 activities:

1. **Plan for One-Time Software Setup and Training**
2. **Prepare for Ongoing Support Needs**
3. **Consider Additional Support as the Software Scope Expands**

Support activities will differ for software that is primarily "customer facing" compared to that which is "agency" facing, and the needs for each of these types of software are described and contrasted.

A series of post-procurement, one-time activities will be necessary in order to make the software operational, namely (1) software deployment, (2) software configuration, and (3) training of the agency's staff in how to use the software. Once these activities have been accomplished, it is essential that the agency make a full commitment to effective use of the software on an ongoing basis. What that means in practical terms is described, in particular the importance of incorporating software support priorities into the agency's planning and budgeting process. This is directly relevant to the ability of the agency to obtain major upgrades of the software and/or additional modules, which is typically desirable.

This chapter, and the Guidebook, conclude with a discussion of how opportunities to expand the scope of an agency's activities are likely to have implications for its software needs and the factors that they need to be aware of as they assess how to respond to such possibilities.

Introduction and Background Information

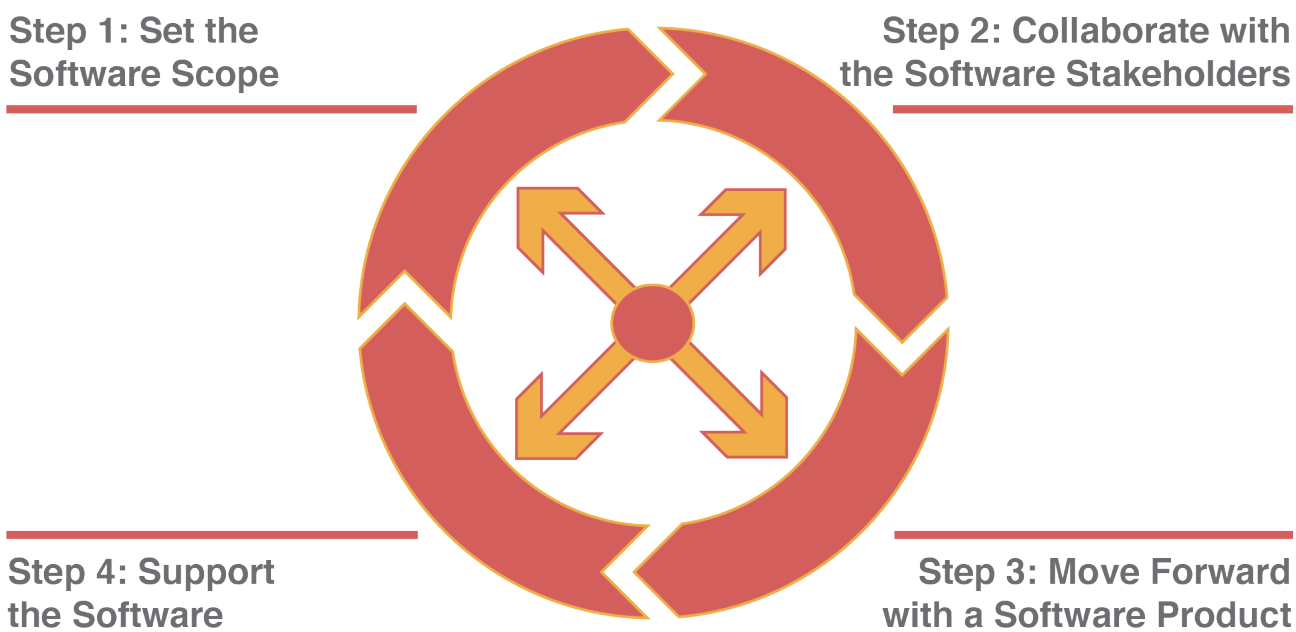
Guidebook Purpose and Structure

Improving the passenger experience and operating community transit service more efficiently often involves utilizing software-based tools. During the past decade, more and more software options have arisen to assist small transit agencies with their operations and passenger interactions. In some cases, agencies adopt new software in order to upgrade from older software, but in other cases, agencies adopt software to replace so-called "pen and paper" work tasks altogether. In either case, the software may lead to—and/or require—significant internal agency adjustments. Not only will staff need to be educated on how to use the software, but the agency may also need to make important internal changes to accommodate entirely new day-to-day work tasks. In this way, software can have a ripple effect into transit agency operations that may not be entirely clear from the outset.

Additionally, the application of new software at a transit agency can work in two directions. On one hand, existing services and their operations may have software applied to them, in order to increase efficiency, bolster productivity, and improve the passenger experience. On the other hand, software may lead to an agency reimagining how it serves its passengers, since some software increases the viability of modes and digital services that may be entirely new to an agency. Software has the potential to fundamentally transform transit agency services.

In short, a holistic process of software adoption is needed, one that takes all critical factors into account and helps transit agencies navigate unknowns and uncertainty. New software adoption has the potential to range from a relatively simple undertaking to an extremely complex one. While the details of an agency's individual software adoption process are unique, specific to the intricacies of each agency applying it, the overall structure of the process can be generalized and broken down into a set of steps to follow. This Guidebook provides a four-step process to move from the initial stages of software consideration to later steps involving set-up, operations, and maintenance. While all the details provided within the Guidebook may not be needed for simple cases, they can serve as a checklist of sorts to ensure an agency has considered key aspects.

The four steps are shown below as a cycle of steps that often occur chronologically in the order shown, but do not necessarily follow such a sequence in all cases. For example, while in the midst of Step 3, an agency may need to revisit some of the details associated with Step 1. Adopting new software is a process that is not always linear, and can often involve some surprises. By reading this Guidebook and applying its information, small transit agency staff will be better prepared to understand the full range of activities involved in software adoption. They will be better able to anticipate challenges and address unknowns; increased awareness will contribute to a more successful software adoption outcome.



Guidebook Focus Areas and Software Types

This Guidebook is targeted primarily at small transit agencies, but medium and larger transit agencies may find it useful as well. Further, it is focused on small transit agencies adopting an already existing "off-the-shelf" software application (i.e., an existing product on the market) and does not include the level of detail needed to guide the creation of a new, customized software application. The Guidebook focuses on specific types of software that are common for small transit agencies, as shown below. Additional software types are explained in further detail in the "Software Functional Types for Small Transit Systems" section of Chapter 3.

- **Trip planning** – Trip planning software enables a customer to view their transportation options online, typically across most or all available choices such as fixed-route transit, demand-responsive transit, bike sharing, walking, and other modes that may be available in an area. This software type enables a customer to quickly compare their options across multiple factors such as trip duration, departure/arrival time, cost, and other factors. More advanced versions of this type of software support "trip chaining" or combining multiple legs of different modes in a single trip (e.g., bike to fixed-route transit or demand-responsive transit to fixed-route transit). This type of software is typically accessed through a website or mobile app. An example of trip planning software is available in N-CATT's "Promising Practices Guidebook: Transit Technology Adoption"—the Go Vermont! Trip Planner.[1] This type of software is a core component of so-called Mobility as a Service (MaaS) applications.

- **Trip booking and scheduling** – Trip booking software enables a customer to book a ride on a transit service and to be provided with information on when the vehicle will pick them up. The trip booking process may be done directly by the customer via a web-based or smartphone app-based application, or may be accomplished by an agent on behalf of the customer using an on-line booking application. Trip booking is very common for demand-responsive transit, and is often directly connected to trip scheduling. Less commonly, fixed-route trips are booked, typically to ensure that riders have an available seat or to control capacity utilization on a vehicle. During the COVID-19 pandemic, interest in trip booking for fixed-route transit became more common as a tactic to increase safety by controlling the number of passengers and seating arrangements. In some cases, trip booking software is directly connected to trip planning software (i.e., customer reviews options via a trip planner and then books the trip seamlessly). An example of trip booking software is available in N-CATT's "Promising Practices Guidebook: Transit Technology Adoption"—the Michigan Ride Paratransit app.[2]

  Vehicle scheduling is a fundamental element of all transportation services, and takes different forms for fixed-route transit and demand-responsive transit, with corresponding differences in the types of software required. For small fixed-route transit operations, the vehicle trips to be scheduled are those on a fixed set of routes with pre-determined timetables for the vehicles. Such schedules can often be managed in spreadsheets or very basic software that also includes driver scheduling.

  Scheduling software for demand-responsive transit is quite different, and typically works in conjunction with trip booking. The scheduling system is invoked when a customer wants to book a trip, and it determines when that trip request can be accommodated based on commitments already made to other customers. The trip scheduling software manages the capacity of the vehicle fleet used in the service. It determines the routing

[1] National Center for Applied Transit Technology. Promising Practices Guide, pp.32-34. Available at: https://n-catt.org/tech-university/promising-practices-guide/ as of February 10, 2021.

[2] National Center for Applied Transit Technology. Promising Practices Guide, pp.20-22. Available at: https://n-catt.org/tech-university/promising-practices-guide/ as of February 10, 2021.

of each vehicle and how customer pick-ups and drop-offs are sequenced on the vehicle, while also ensuring that customers do not stay onboard for an excessive period of time. The scheduling system is at the heart of the software that manages a demand-responsive transit system. There are many levels of sophistication for such scheduling systems, from those that do little more than organize trips for a dispatcher to schedule manually to those that are fully automated and require little or no human intervention.

- **Trip payment** – Trip payment software enables a customer to pay for their trip online. Such software is tied to multiple payment methods such as credit and debit cards, but may also have options for "unbanked" customers. Such customers may be provided with an option to load value in advance through cash or other payment methods, and then this value is connected to their online account. In some cases, trip payment software is directly connected to trip planning and/or trip booking software (i.e., customer reviews options via a trip planner, books the preferred trip, and then pays in advance). Trip payment software is typically accessed through a website or mobile app. An example of trip payment software is available in N-CATT's "Promising Practices Guidebook: Transit Technology Adoption"—the EZfare implementation for seven Ohio transit agencies.[3]

Software Types and Related Concepts

The software types explained above are presented in a way that connects with concepts such as Mobility as a Service (MaaS) and One-Call/One-Click Systems. N-CATT has published a resource on MaaS, "Mobility as a Service - Now and In the Future," that serves as a reference.[4] In addition, the National Center for Mobility Management (NCMM) has an online One-Call/One-Click Resource Center with additional information.[5]

[3] National Center for Applied Transit Technology. Promising Practices Guide, pp.39-42. Available at: https://n-catt.org/tech-university/promising-practices-guide/ as of February 10, 2021.
[4] National Center for Applied Transit Technology. Mobility as a Service: Now and in the Future. Available at: https://n-catt.org/tech-university/mobility-as-a-service-now-and-in-the-future/ as of February 10, 2021.
[5] National Center for Mobility Management. One Call One Click Resource Center. Available at: https://nationalcenterformobilitymanagement.org/one-call-one-click-resource-center/ as of February 10, 2021.

Software for Demand-responsive Transit Operations

Demand-responsive transit (DRT) is a transit mode that does not operate on a fixed schedule. Instead, it involves customers booking their trips in advance; the booked trips in turn feed into trip scheduling software that is used behind-the-scenes at transit agencies to generate the routing of the vehicles to pick-up and drop-off passengers (sometimes called "vehicle tours"), as discussed previously.

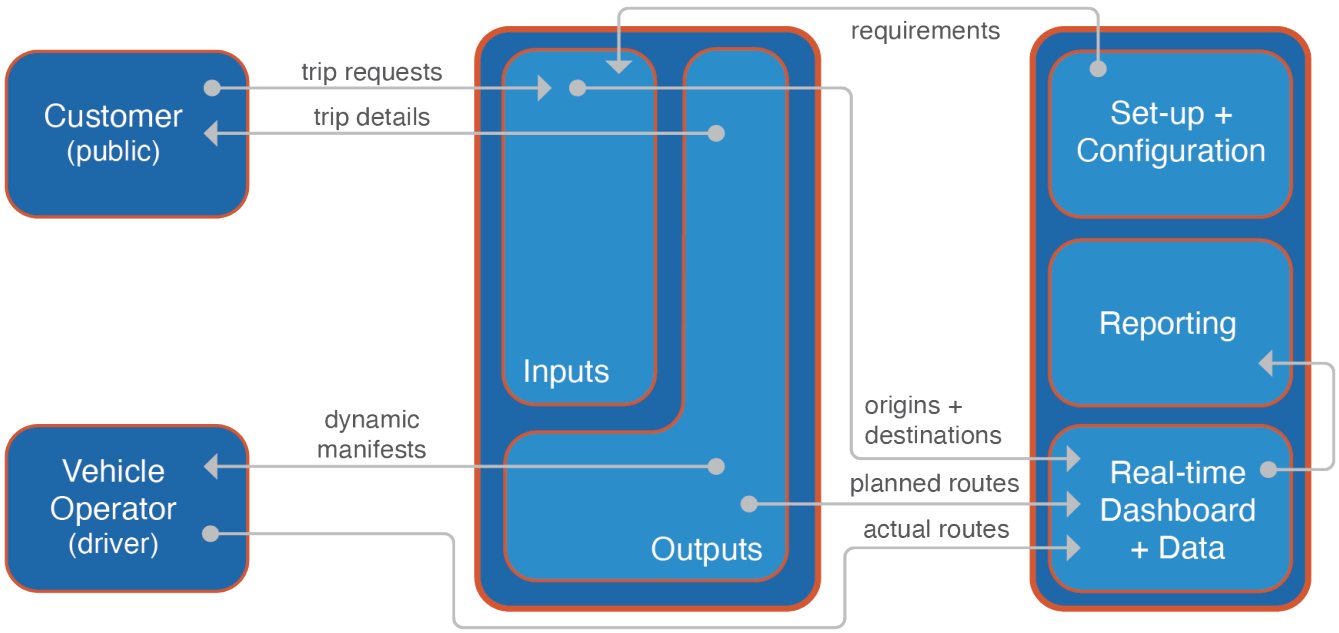
DRT is part of a broader concept of flexible transit services which includes other service models, such as deviated fixed-route service, a hybrid service combining fixed routes and DRT. DRT trips can be booked anytime from several days to a few minutes in advance, depending on how a DRT service is organized. Trips that are booked and taken immediately are sometimes referred to as "on-demand transit" to differentiate from DRT service that requires booking well in advance of service delivery, such as the day prior for most ADA paratransit services. On-demand transit also goes by other names such as microtransit.

DRT services can be provided either directly by agencies using their own staff and drivers to operate the service, or by third-party operations contractors who are fully responsible for the operations components of the service, in some cases providing the vehicles as well. There are several national scale DRT operations contractors (e.g., First Transit, MV Transportation, TransDev) as well as regional scale and local contractors. Some of the latter may also operate taxi services and/or a variety of transportation services under contract to public and private organizations. When an operations contractor is used by a small transit agency for DRT service, it may be a turnkey situation in which the contractor provides the entire service delivery capabilities including all of the DRT software (the vehicles can still be provided by the agency rather than the contractor with a turnkey operation). In other turnkey cases, the agency may provide the DRT software and the contractor must have competency in using a specific DRT software system. If the agency itself operates the DRT service, it will be responsible for the DRT software system. This Guidebook focuses only on software for agency-provided DRT services and for those contracted DRT services in which the agency provides the software system.

DRT ranges in who it serves and where pick-ups and drop-offs occur. It can serve the general population or specific groups such as older adults and riders eligible for certain services (e.g., health care-related trips). The service may be provided directly from one address to another address, such as with curb-to-curb or door-to-door service, but it can also involve pick-ups and drop-offs at street corners or designated locations such as a town center. Further, DRT service may be designed to facilitate pick-ups and drop-offs at fixed-route service connections in an effort for first and last mile connectivity. Some DRT software includes the functionality to support such service variations as deviated fixed-route service (i.e., facilitates the hybrid approach), pick-ups/drop-offs at designated locations such as street corners (i.e., in order to increase service efficiency and productivity), and other operational variations.

While DRT has been around since the 1970's and exists in several hundred communities in the country (for decades in many of these locations), developments in technology have made it more viable and customer friendly. The key developments include the Global Positioning System (GPS), mobile devices, and cloud computing. GPS enables the current location of any moving item, a person or a vehicle, to be known. When GPS is contained within mobile devices that are increasingly ubiquitous for individual passengers and on-board transit vehicles, the infrastructure is in place for real-time information on the location of passengers and vehicles. With cloud computing, software systems no longer need to operate on computers located on an agency's premises and be supervised by an agency's staff (or external contractors). Instead, software becomes a "utility" that one connects to via the Internet; an organization can receive its benefits without concerning itself with the complications of a technology infrastructure. A new generation of DRT/on-demand transit software is built on top of these infrastructural and technological advances, which makes it possible to connect trip booking directly from the passenger and trip scheduling across the vehicle fleet and vehicle operators—all in real time. Equally important, this software is relatively affordable and requires no long-term investment in equipment and staff by the agency.

Since on-demand transit does not operate on a fixed schedule, the software that supports it requires a number of features to facilitate its operations. Common components are shown below. On-demand transit software is typically purchased as a package that includes most or all of these components. These components are designed to work together in support of on-demand operations—facing the multiple software users including passengers, drivers, and administrative staff.



Graphic source: Atlanta Regional Commission, internal reference document

Software for Multiple Agencies

Software may be adopted by one agency, but in some cases, multiple agencies adopt a software application together. Trip planning, for example, is often taken on as a multi-agency software adoption effort. The purpose of trip planning is for the customer to gain an understanding of all their transportation options, often across large multi-jurisdictional rural or urban areas. Such areas connect homes and jobs, shops and medical facilities, and people travel across them to get what they need—often unaware of how many jurisdictions they cross. Therefore, all transportation modes to all key destinations should be included in the trip planning software and customer app. In order for such a software application to be effective, data will need to be sourced from multiple organizations providing fixed-route transit, DRT, micromobility options such as bikeshare and electric scooter share, and other transportation options. All geographic areas that are commonly traveled across should be included—often requiring several jurisdictions and agencies to be involved. The software system itself becomes the "easy" part of acquiring the capabilities of the software, since without comprehensive data the system has limited value.

Trip booking and scheduling software is less often selected as a common software for multiple agencies. On-demand transit operations software has multiple software components operating as one platform, and by its nature, is highly complex software that each agency will commonly want to select for themselves. The software, with its scheduling and administrative components, is intricately woven with how on-demand operations work on the ground. In contrast, trip planning software reflects services offered and typically does not directly impact operations—making it more able to support the needs of multiple agencies at once.

Trip payment, like trip planning, often needs to be facilitated across multiple jurisdictions. Multi-jurisdictional rural and urban areas sometimes have regional transit fare payment structures that streamline and normalize fares, for instance so that passengers don't need to pay two agencies to take a cross-jurisdictional trip. In addition to payment structures, there may also be related regional policies such as defining who rides for free. As with trip planning, having a common trip payment software does not necessarily impact operations on the ground.

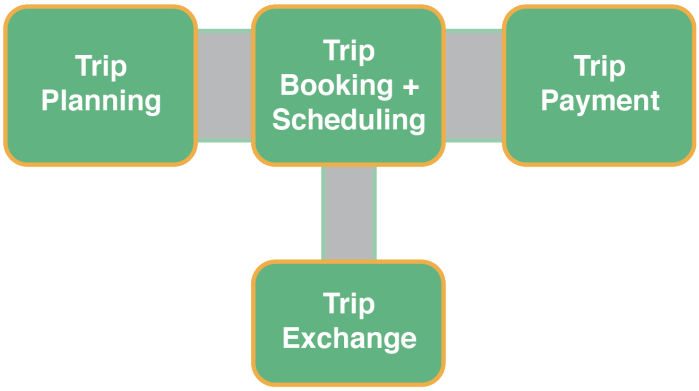
Further, some areas have regional fare media (e.g., RFID cards) that enable payment through a single medium across multiple transit agencies. As mobile devices have become more ubiquitous, mobile payment has been leveraged as a regional fare medium. Agencies that never had RFID cards can move directly into mobile payment, transitioning into a more modern fare collection medium. It is possible for trip payment software to be selected for use by multiple agencies across a broader, connected geographic area. However, it is important to not equate the usage of a common trip payment software with fully integrated regional fare structures or fare policy. Some regional agencies decide to use the same mobile payment software because it is easier for them to procure or because they know it is better for the customer to have a single app. However, in such a situation, there is a range of how integrated the payment structures might be behind the scenes. There may be no integration at all (i.e., simply two agencies on the same software), full integration (e.g., integrated fare structures and fare policy), or something in between.

Software for Integrated DRT Service

Some transit agencies prioritize DRT operations that are integrated with other agencies, often because there is an awareness that passengers are eligible for (i.e., able to use) multiple DRT services. At any point in time, one agency over another may be better positioned to serve the passenger due to their planned schedules, real-time vehicle locations, and number of passengers. There are a few ways that organizations can go about integration including brokerage (i.e., having a central authority that decides which provider will handle a given trip), a common provider (i.e., a single provider that handles trips for multiple eligibility-determined services), and by "exchanging" trips.

With a brokerage or a common provider, the organization providing or allocating the trips will adopt its own DRT operations software, and due to their central role, theirs may be the only DRT operations software in use for the system. In contrast, "exchanging" trips involves leveraging software in order to comingle passengers that are eligible for multiple services handled across multiple providers. In this situation, each provider has their own DRT operations software, and exchanging trips requires using a separate software designed to interact between the multiple DRT operations platforms. In general, a "trip exchange" software involves keeping track of all the services for which

a passenger is eligible, the ability for providers to "post" a requested trip for another provider to "claim" (i.e., matching trip requests with the provider best positioned to handle them), and cost reconciliation to track the financial details of each trip (i.e., allocating the cost among providers and funding categories). In essence, trip exchange software is a secondary software that works between multiple software platforms for DRT operations.



Since DRT services often have adjacent and overlapping service areas, the "comingling" of passengers who are eligible for multiple services can help make the most of a transportation delivery system that has multiple providers operating in the same service area. Applying trip exchange software will not change the overall structure of the way trips are handled, often across a patchwork of providers and services, but it can increase the efficiency of such systems by improving the balance between supply and demand. It can also help improve the passenger's experience in trying to book a trip, getting denied, trying to book with another provider, and so forth. With an exchange, although a passenger attempts to book with an initial provider that cannot handle it, if another provider ends up accepting the trip, they will receive communications from the initial provider explaining the changes—saving them time and confusion in the process.

A trip exchange can support integrated DRT operations for different situations, such as between multiple transit agencies (e.g., to comingle ADA paratransit clients eligible for multiple, adjacent ADA paratransit services), between a transit agency and a human services transportation or non-emergency medical transportation provider (e.g., to comingle clients eligible for multiple services), and other situations.

The "illustrative project" provided in Chapter 4 explains how a trip exchange software works in further detail.

Guidebook "Quick Content" and Worksheets

The Guidebook "quick content," at the end of each chapter, includes key takeaways and illustrative projects. Key takeaways are a set of activities that help the reader quickly understand what steps are proposed for the software adoption process. Illustrative projects provide context around the topics mentioned in the chapter; they are real-world examples of how an agency or organization has tackled a specific activity as they navigated the software adoption process. For readers interested in gaining further information beyond the executive summary, but not necessarily planning to read the Guidebook in its entirety, reviewing the quick content is the next best step.

If, after reviewing the Guidebook's content, a reader is interested in applying the information to their particular situation, a set of worksheets is provided at the end of the Guidebook. The worksheets walk a reader through the activities mentioned in each chapter, with clear guidance on how to apply the information. For some agencies, completing the worksheets individually or during a collaborative session could be the next step in better understanding how their own software adoption process could work. The worksheets help Guidebook users organize their thoughts, pinpoint gap areas, and plan for upcoming efforts.

About N-CATT

The National Center for Applied Transit Technology (N-CATT) is a technical assistance center funded through a cooperative agreement with the United States Department of Transportation's Federal Transit Administration (FTA). Operated by the Community Transportation Association of America (CTAA), the mission of N-CATT is to provide small-urban, rural and tribal transit agencies with practical, replicable resources that help them apply technological solutions and innovations. Among its activities, N-CATT produces a series of white papers, technical reports such as this document, and other resources, all of which can be accessed on-line at https://n-catt.org.

About this Document

This document was prepared for CTAA by DemandTrans and Civic Sphere as part of the N-CATT cooperative agreement between CTAA and FTA. Primary authors were Dr. Roger Teal of DemandTrans and Janae Futrell of Civic Sphere. Opinions expressed or implied in this document are those of the authors. Nothing in this document is to be interpreted as position, policy or guidance from the United States Government. Incidental use of companies' names or the names of their products is made solely to facilitate discussion and should not be regarded as recommendations or endorsements.

Chapter 1
Step 1: Set the Software Scope

This chapter breaks down the process of setting the software scope into three activities. For these activities, as well as all the other activities explained in this Guidebook, an agency should identify a primary staff member who is responsible for the entire software adoption process from start to finish—a software adoption process lead. They would lead all the activities, from Step 1 through Step 4. Some software adoption efforts end up having difficulties because the agency was unaware that certain activities needed to be completed, or because there was no single point of contact. Having at least one staff member who understands all the parts and how they should come together will reduce the risk of an important activity falling through the cracks.

The lead does not need to come from a software development or information technology background. The most important skill sets and knowledge the lead should have include:

- Transit operations knowledge – They should have a solid understanding of how the software would be used by the agency staff and by the public, as well as which groups in particular would use the software.
- Software benefits awareness – They should be able to clearly communicate with staff and the public about why the software would be useful and how it would improve operations and/or the user experience.
- Organizational skills – They should know the basics of project management, how to plan for large scale projects with many parts, and understand they are responsible for bringing all the parts together. They should be someone who can work with a wide range of people to reach a common goal.

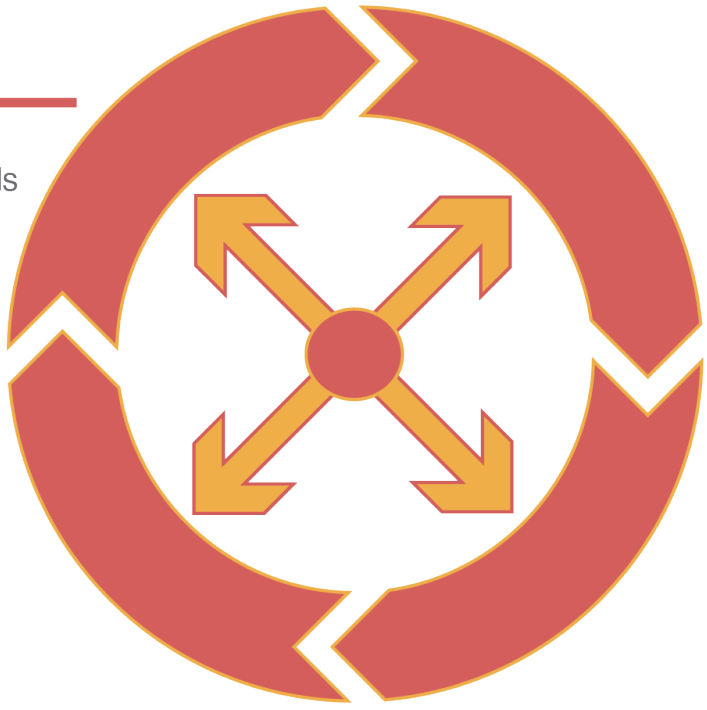
"Key takeaways" are provided at the end of each chapter to help orient the software adoption process lead toward the most critical activities.

Step 1a. Clarify the Software's Purpose

Clarifying the purpose of the software involves identifying which software type is needed. In general, when beginning the process of identifying which software type is the most applicable to a transit agency's needs, two situations are common. For some internal needs, it is clear to staff which software type is required, or if multiple types are required. For example, if an agency is starting a demand-responsive service for the first time, and they have decided that it will be agency-operated

Step 1: Set the Software Scope

1a. Clarify the software's purpose
1b. Identify general software connectivity needs
1c. Anticipate resources



as opposed to a fully turnkey service, they are aware they will need a DRT software system. In this situation, the software's purpose is clear; the agency needs a DRT technology platform in order to operate new DRT service.

For other internal needs, it may be unclear what is needed exactly, and in such cases, the agency may need to first explore their options. For reference, the software types of focus for this Guidebook are shown below. For further information, refer to the "Guidebook Focus Areas and Software Types" section of the Guidebook's Introduction and Background Information as well as the "Software Functional Types for Small Transit Systems" section of Chapter 3.

In this situation, an agency may want to consider ways of pinpointing the type, or types of software that are most needed. The N-CATT white paper, a "Framework for Making Successful Technology Decisions," provides detailed instructions on ways to navigate this situation.[6] The Framework encourages readers to think about technology needs holistically and consider the full realm of what might be needed, even if that means considering multiple types of software within a broader technology portfolio. Since the Guidebook focuses on a narrower category of technology, software only, the results of the suggested Framework process may expand beyond software. In fact, such results could be helpful in the longer term, because the agency would have greater clarity into all the types of technology that are needed—and how they should work together to suit the agency's needs.

The Framework encourages a few methods in the first phase of activity,[7] such as:

- Defining and ranking problems – It is suggested to focus first on what problems the agency faces in the form of identifying "pain points."
- Creating problem statements – This takes the problems from general observations to a more specific statement that is targeted. Each statement should connect to the transit agency's mission and avoid naming a solution.
- Evaluating problem statements based on risk – This method involves assessing the potential level of impact of the problem on core needs of the transit agency, such as service reliability, as well as its likelihood of happening. The assessed impact level and likelihood can then be mapped onto a risk management matrix for comparison.

In the second phase of activity,[8] solutions are identified. This phase can be connected to the software types of focus for the Guidebook in order to match software types, as potential partial or complete solutions, to the problems. It is possible that the activities mentioned in the Framework result in a set of technology-related solutions that don't involve any software, but if the need for specific types of software is revealed, the Framework information can be used within this Guidebook step to clarify the purpose of the software product.

If an agency applies these methods early on in the software adoption effort, they may have a solid base for not only a complete technology portfolio, but also technology plans or roadmaps that connect elements of the portfolio with time-based phases to break down when certain elements may be implemented.

Step 1b. Identify General Software Connectivity Needs

If, while clarifying the purpose of the software, an agency determines that it will need to adopt more than one type of software, then the connectivity needs of the new software will also need to be identified. Further, new software always connects into an agency's broader software ecosystem, and exploring this as well can help with the adoption of new software. An agency should make sure it understands all existing software that may have some connection, however limited, with the new software. At the same time, it should look ahead to

[6] National Center for Applied Transit Technology. A Framework for Making Successful Technology Decisions. Available at: https://n-catt.org/tech-university/a-framework-for-making-successful-technology-decisions/ as of February 10, 2021.

[7] National Center for Applied Transit Technology. A Framework for Making Successful Technology Decisions, pp.13-18. Available at: https://n-catt.org/tech-university/a-framework-for-making-successful-technology-decisions/ as of February 10, 2021.

[8] National Center for Applied Transit Technology. A Framework for Making Successful Technology Decisions, pp.18-30. Available at: https://n-catt.org/tech-university/a-framework-for-making-successful-technology-decisions/ as of February 10, 2021.

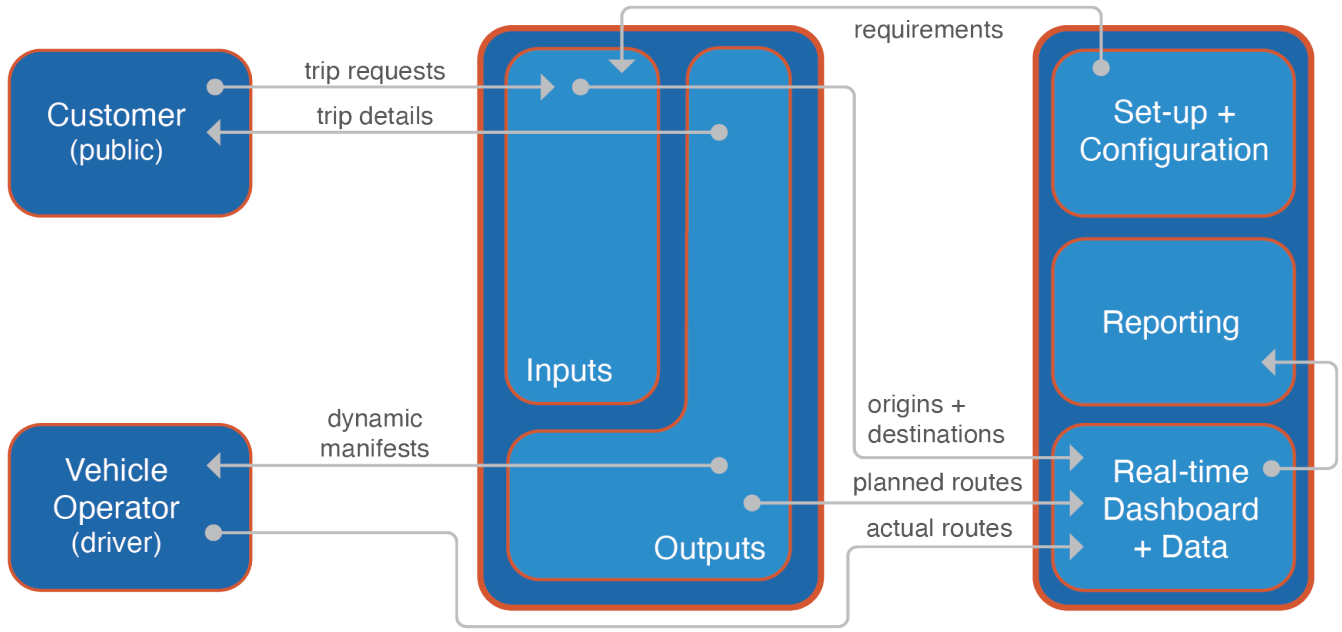
future planned software that may also have some connection.

For the purposes of this Guidebook, a software "connection" is defined broadly. On one end of the spectrum, a connection could mean that a staff member uses both types of software in their daily work. Perhaps although they use both, they don't necessarily need them to exchange data or directly connect. Nonetheless, in this example, it should be noted that the two software platforms contribute to common, or related, work tasks. On the other end of the spectrum, a connection means that there is a direct relationship between the two, such that data must be exchanged between them or that they perform functions directly connected to each other. This side of the spectrum involves software that is interoperable.

Conceptually, the term "connectivity" could also be used in reference to Internet connectivity, which is facilitated by connecting mobile devices (e.g., smart phones and tablets) to the Internet via wireless location-based routers (i.e., wifi) and via cellular data that covers wide geographic areas. For the purposes of this Guidebook chapter, "connectivity" refers to *software* connectivity, not *Internet* connectivity.

An example of interoperable software, on-demand transit operations software, is shown below. This software, or software platform, is comprised of multiple pieces of software, each with a high level of interoperability with one another. There are many connectivity needs that exist. The customer inputs their trip request into the customer app, which is sent to the scheduling software; this is one connection that is needed. Once a manifest is created, the details are sent to the vehicle operator, who is using an app for navigation purposes; this is another key connection. All the information on the trip request and final trip delivery details are available as data to be leveraged in the reporting module or the data dashboard—an additional needed connection. In this way, all the components are designed as parts of a whole that work together to achieve a variety of critical results.

In the example just cited, the software vendor provides a comprehensive, integrated solution to all of the needs of an on-demand transit service, as would typically be the case of such software that has become available in the past 2 or 3 years. But it can also be the case that an agency has older software that is less comprehensive and integrated, and needs to be augmented in certain ways, or has a much simpler DRT service that is perceived not to require the same level of comprehensiveness and integration as what is now the state of the art. Even in such cases, it is important for the agency to have a solid understanding of the connection needs of its



Graphic source: Atlanta Regional Commission, internal reference document

software—current and future—so that if the agency does pursue additional software products then it will consider them functioning well with one another. If it does not think comprehensively about how the different software elements should work together, the agency staff may find themselves in a situation where they continue to need to handle things manually, costing the agency time and money—contributing to a lack of productivity and efficiency in its internal operations.

While software interoperability can help an agency with its services internally, it also impacts passengers. Mobility as a Service (MaaS), for example, stresses the importance of making trip planning, booking, and payment easier on the passenger through interoperability. In this sense, interoperability can mean that one app supports multiple functions seamlessly such as trip planning, booking, and payment—each function would be interoperable with the other. On the other hand, some level of interoperability could potentially be accomplished through multiple apps or software that are designed to work together.

Interoperability between separate software or apps can either be custom-built or achieved by using an intermediate method such as an application programming interface (API).[9] When a software developer knows in advance that two software products should interoperate, it is possible to build in what is needed to facilitate the interoperability as the software is being designed, written, and tested. However, it is far more typical for a software to instead have an API built that works as a sort of appendage to the software in support of connecting that software with another software. APIs have the potential to facilitate the connection of one software product with countless other software products.

By listing all of the existing, new, and future planned software and thinking through the spectrum of connections that should ideally exist between them, an agency can clarify its connectivity needs. Within Step 1, only a general connectivity understanding is needed. Software interoperability and APIs are covered in further detail under "Inter-Operable Software Considerations: A Short Discourse" within the "Software Functional Types for Small Transit Systems" section of Chapter 3.

[9] InfoWorld. What is an API? Application programming interfaces explained. Available at: https://www.infoworld.com/article/3269878/what-is-an-api-application-programming-interfaces-explained.html as of February 10, 2021.

Step 1c. Anticipate Resources to Apply to Software Adoption

Software adoption can benefit from an early look into an agency's resources to consider what can be brought to bear on various aspects of the adoption process. Resources include, at a minimum, financial resources, staff resources, assets, and collaborator resources.

- Financial resources include sources such as dedicated funding and grants that can support purchasing a new software and potential associated tasks such as software training.
- Staff resources should be explored and can include those who might oversee the deployment and maintenance of the software as well as those who have related skill sets such as data creation. A significant staff resource to any effort is the software adoption process lead, since they will be responsible for leading the activities in Steps 1-4.
- Assets include existing resources an agency has that can be useful for the new software, such as current software or databases to leverage, or servers and other IT infrastructure that might be needed.
- Collaborator resources involve those that may come from a partner organization and could include financial resources, staff resources, or assets. Perhaps the collaborator benefits indirectly from the new software and would like to make an in-kind contribution to the effort.

An agency can create an inventory of all its anticipated resources early on in the software adoption effort in order to be prepared for later Guidebook steps. Chapter 3 includes details of the procurement process and the deployment of financial resources. Chapter 4 refers to resource-dependent tasks such as training staff members and maintaining the software.

Key Takeaways

- Clarify the software's purpose by connecting the transit agency's needs with the corresponding software type or types. For situations when it is unclear which software type is needed, apply the methods provided in the N-CATT white paper, a "Framework for Making Successful Technology Decisions," to explore an agency's technology portfolio more broadly.
- Identify general connectivity needs by listing all of the existing and future planned software that will have a relationship, even a loose one, with the new software type or types. The details of the connections are not needed during Step 1, only the understanding that some sort of connection should exist.
- Anticipate resources to apply to software adoption by creating an inventory of all an agency's potential resources, within the agency and from partner organizations.

Illustrative Project

LYNX is the transit agency for central Florida, serving counties including Orange, Seminole, and Osceola as well as limited service to Polk County. Orlando is included in the LYNX service area (estimated population of 287,442), as are municipalities such as Apopka (estimated population of 53,447), Oviedo (estimated population of 41,860), Sanford (estimated population of 61,448), and St. Cloud (estimated population of 54,579).[10] LYNX has a unique story behind a number of its software platforms, from the standpoint of connectivity between the platforms as well as innovative ways of anticipating resources. LYNX has platforms that support trip planning, trip booking/scheduling, and trip payment.

The first app made available to the public, in March 2016, is an online trip booking platform called WebAccess for LYNX's Americans with Disabilities Act (ADA) paratransit service, Access LYNX. One online booking platform supports central Florida users of both the ADA paratransit service and the Florida-based Transportation Disadvantaged program, a "coordinated state-wide effort which groups riders together for a shared ride service. Transportation services are available in all 67 Florida counties for those who are eligible and have no access to transportation. Federal, State and Local agencies join together to provide necessary transportation to medical

[10] United States Census Bureau. Quick Facts. Available at: https://www.census.gov/quickfacts/ as of February 10, 2021.

appointments, employment, educational and other life sustaining services."[11] The platform provides the same service for users regardless of the funding source, and the funding source for each trip is reflected in the user's account (i.e., ADA or Transportation Disadvantaged). In order to accurately allocate trip payment and cost, the trip-specific funding source is noted behind the scenes when the user books their trip and completes the billing process. This is particularly important when multiple funding sources are involved in a single trip booking platform, avoiding administrative headaches and financial misallocations. Further, the platform provides each user with their own account, including a unique identifier (i.e., a client identification number), so that they can keep track of their own activity, and LYNX can see the big picture of activity among all users. Users of Access LYNX are still able to call the LYNX "Mobility Services" call center to book trips. The app helps the public save time in the booking process, but has also significantly reduced LYNX's call center call volume as more bookings are handled online.

Image source: LYNX

The second app made available to the public, in March 2019, is an online and mobile trip payment platform called PawPass. Online and mobile payments are possible on all of LYNX's services, including ADA paratransit and Transportation Disadvantaged services. Since online booking was already possible for these services through WebAccess, LYNX got to work figuring out how the users could also pay for their trips on their mobile devices prior to taking the trip. The system in place currently is as follows; once the trip is booked WebAccess, the user shifts to using the PawPass app to pay for the trip. Once on board the vehicle, they can show the driver their active ticket. The unique identifier for the user in the WebAccess app is the same in PawPass. When getting started in PawPass, LYNX requires the user to enter their Access Lynx ID (i.e., connected to the unique identifier/client identification number) to request approval. Thus, although there are two apps and technically two user accounts, the user accounts are linked via the unique identifier and can then communicate information consistently regarding a specific user. In addition, users who book trips via LYNX's "Mobility Services" call center can also pay online through PawPass. The user experience is much improved by having the opportunity to handle both booking and payment transactions online, and the fact that users need to use two apps in order to accomplish

[11] Florida Commission for the Transportation Disadvantaged. About us. Available at: https://ctd.fdot.gov/aboutus.htm as of February 10, 2021.

the tasks is made less impactful since the two apps are able to communicate key information between one another.

The third app made available to the public, in November 2019, is a trip planner called LYNX Connects. This trip planner is focused on providing information on demand-responsive options. While many agencies begin with a trip planner first and then add on other platforms for booking and payment, LYNX went in a less-common but equally as relevant order. They tackled the online trip booking needs of users of the ADA paratransit and Transportation Disadvantaged services first, and in the meantime, they incorporated their fixed-route transit service details through LYNX's General Transit Feed Specification (GTFS) data into the Google Maps trip planner.

LYNX was the recipient of a Mobility Services for All Americans (MSAA) grant in 2005, which enabled the agency to design a concept called the Model Orlando Regionally Efficient (MORE) Travel Management Coordination Center (TMCC) in order to better coordinate transportation and technology services in its area. The 2005 round of funding is considered phase 1 within the wider scope of MSAA grants provided, with phases 2 and 3 taking place in 2009 and in 2015, respectively. When LYNX did not receive the phase 2 funding in 2009 to help implement the design created during phase 1, LYNX staff members decided to pursue a lighter version of the TMCC that could be funded in-house. This lighter version consisted of the WebAccess and PawPass platforms. The LYNX Connects trip planner was funded through a Veterans Transportation and Community Living Initiative (VTCLI) grant.

LYNX's story is a helpful illustration of how there is rarely a "wrong" way to approach improving digital services for the public. An agency should start wherever it makes the most sense for them and then plan for making later improvements as they go. Sometimes the end point is not 100% clear, and some agencies continue to clarify their vision—or refine their past ideas—as they implement and learn through their own process.

Although these platforms are custom-designed, and are not off-the-shelf products, the lessons to take from LYNX's process of building their software portfolio strategically over time, in an integrated and connected way, are useful for all transit agencies of any size. While software products are sometimes seen as complete solutions that automatically connect with each other with relative ease, transit agencies working on such needs know that the path rarely runs as smoothly as one would hope. It is often a long-term journey of many steps—each one pushing the effort a bit further in the right direction. Eventually, an agency can step back and look at how well all the pieces plug into one another. Not every software connection needs to be fully "seamless" and interoperable; a great deal of customer benefits can be derived from pieces that connect reasonably well and lay the groundwork for potential improvements down the line. Further, a patchwork of funding and other resources can support such efforts. LYNX leveraged not only grants, VTCLI and MSAA, but also in-house funds to fully realize their vision over 15 years from beginning to current state. With the common hurdles all agencies face—staff changes, shifting institutional priorities, and other potential setbacks—LYNX has demonstrated a long-term commitment to building software solutions piece by piece over many years.

More details about this project are available on the National Center for Mobility Management's (NCMM) One-Call/One-Click Resource Center, on the central Florida spotlight project page.

Chapter 2
Step 2: Collaborate with the Software Stakeholders

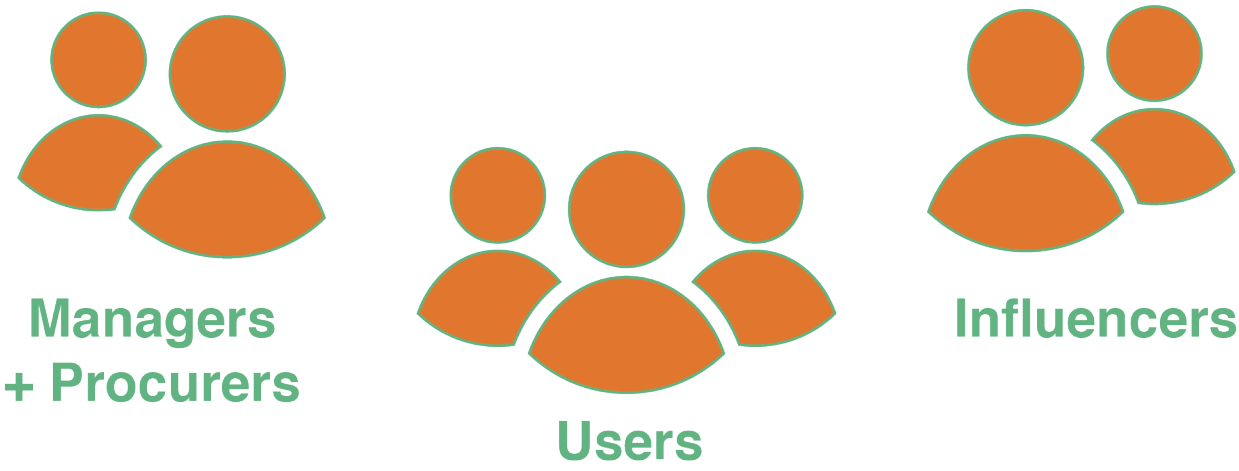
A core aspect of software adoption involves working effectively with stakeholders. Far from an optional step, this aspect requires significant attention early in the adoption process. First, ways to identify which stakeholders to include will be explained. Then, ways to actively involve them will be provided. Although the involvement of stakeholders is proposed to begin after Step 1, it is possible that a stakeholder group should be involved earlier—during Step 1. In fact, for situations mentioned in Step 1, when the software type is initially unknown, a more thorough stakeholder-related effort is suggested to take place during Step 1.

Identifying the full range of stakeholders is important for two reasons. First, it establishes a deeper understanding of who should be included in the decision-making and adoption process, increasing the likelihood that the needs of each group are met. This, in turn, helps the software adoption process to be more effective and run more smoothly. Second, it helps avoid the pitfalls that can occur when software decisions are made on behalf of software users without their direct involvement in the process. Pitfalls that may happen in this case include a lack of use of the software by users or a premature commitment to a software that does not have all the required functionality—either of which may lead to a waste of funds and staff time.

Software stakeholders can generally be broken down into three categories:

1. **Managers and procurers** – The management of a software adoption process involves guiding its direction and ensuring it remains on track. As mentioned in Chapter 1, a staff member who is responsible for the entire software adoption process from start to finish should be identified. This person, the software adoption process lead, would be the primary manager of the effort. Similar to a project management role, the software adoption process lead would ensure that all the parts—the results from Step 1 through Step 4—connect.

   Procurement involves selecting the software in accordance with the agency's requirements, policies, and procedures. Staff involved on procurements tend to break into two types, dedicated procurement staff and subject matter expert(s). While the former ensures that the policies and procedures are adhered to, the latter helps draft the detailed content in procurement materials. The subject matter expert is not necessarily knowledgeable about the software itself, but about the work areas the software will support (e.g., on-demand transit service) and how it will be applied to meet the agency's needs. It is likely that the main subject matter expert involved in the procurement will be the software adoption process lead. Sometimes procurers and managers are also the only direct users of the software, but other times they procure and manage the software on behalf of a larger user group.



2. **Users** – The software user groups encompass everyone who will interact with the software; they should be identified up-front to ensure later steps of the adoption process take the viewpoints of all users into account. For some types of software, the user groups include both members of the public and agency staff. For on-demand transit operations, for example, typical software components include an app for customers (i.e., public users), an app for vehicle operators, and a management console for operations staff. This diversity of software interactions leads to a more complex set of user groups. Each user group will have a different set of requirements based on their needs.

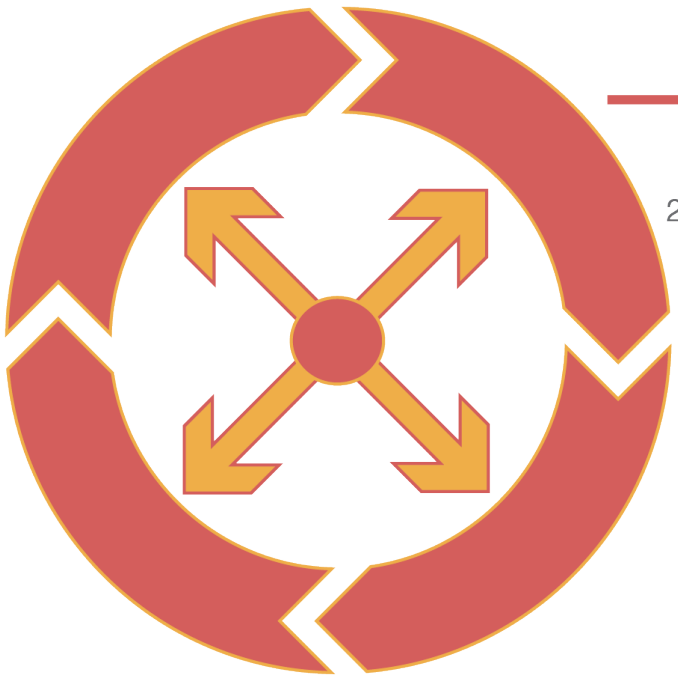
3. **Influencers** – Influencers are stakeholders that won't directly use the software, nor will they be procurers or managers, but they will provide input into the process. For example, it is common in human services transportation to include the advice of groups that represent older adults, individuals with disabilities, and others with specific mobility considerations. Such groups might review the software options from an online accessibility standpoint or check that users can communicate specific types of information regarding accommodations through the booking process (e.g., inclusion of a travel companion).

Some software adoption efforts will have a simpler and shorter list of stakeholders. For example, some efforts will only have procurer and manager stakeholders, and they may also be the only users. For other efforts, all three categories may be involved, and the users may break down into several user groups—adding up to a more complex and longer list of stakeholders. As a general rule, the more complex the software, the more stakeholders will be involved—and the interests of the stakeholders will vary more widely.

Staff at small transit agencies are often wearing many hats and have little available time, so it is critical that the stakeholder involvement aspects of software adoption be handled in an efficient, yet effective, manner. To this end, three key activities are proposed.

Step 2a. Create a Stakeholder Map

As a first step, the stakeholders will be identified through a stakeholder map. The software adoption process lead might complete this step alone or with co-workers as a group brainstorming activity. A stakeholder map can be a list or a graphical sketch that identifies connections (e.g., cases where the "procurer/manager" stakeholders are the same as the "user" stakeholders). Points to consider to help identify the stakeholders for each stakeholder category are shown in the table on the following page.



Step 2: Collaborate with the Software Stakeholders

2a. Create a stakeholder map
2b. Identify key topics for each stakeholder group
2c. Create a tailored information-gathering process to integrate stakeholder findings

Stakeholder Category	Points to Consider
Managers and Procurers	▪ All organizations and known roles/individuals who will be taking part in the management or procurement of the software should be listed. ▪ For management, this would include the software adoption process lead as well as others working on management aspects. For instance, there may be a broader management team involved in the effort, or there could be staff members who will support the software adoption process lead during certain steps due to their specific role or specialized knowledge. ▪ For procurement, this would include dedicated procurement staff (i.e., those who ensure policies and procedures are adhered to) and subject matter experts (i.e., those who help draft the detailed content in procurement materials).
Users	▪ First, the user groups should be identified. The user groups will be different for each software type. For example, for trip planning and trip payment, a key user group would be the customers, members of the public who will use the app. Drilling down further, there could be more nuanced public user groups, such as individuals with disabilities or older adults—this depends entirely on the purpose of the software and the agency's goals for its use. ▪ For trip booking and scheduling, the user groups would align with the software components. Members of the public, as app users and customers, would be a key user group. Various staff members would also be users, since interacting with the software will become a part of their daily work. Typically, the vehicle operators would have an on-board app to help navigate routes, and service operations administrators would have software to assist them with set-up and configuration, reporting, and data analysis. This example is not exhaustive or standard and depends on specifics of the software platform that is needed. ▪ It is common that the manager/procurer stakeholders will be users themselves, but it is not always the case—some software is managed or procured on behalf of users who are entirely different. ▪ After all the user groups are listed, then the list can be further completed with organizations and known roles/individuals at the organizations.
Influencers	▪ Any organization with a role in the effort that has not already been listed should be added. ▪ Typically, such an organization would be included because they will provide input into the process, but it is also possible that they have some other vested interest in the software adoption effort.

Step 2b. Identify Key Topics for Each Stakeholder Group

Each group will have a set of topics that pertain closely to their interests. Points to consider to help identify the key topics for each stakeholder category are shown in the table below.

Stakeholder Category	Points to Consider
Managers and Procurers	▪ Initial topics for managers and procurers relate to the software scope as covered in step 1, including the purpose of the software/software type(s), software connectivity needs, and a resource summary (i.e., available resources, required resources, and resource gaps) for the software adoption effort. ▪ They would lead the selection and procurement process during step 3, with the input of other stakeholders, and would ensure plans are in place for various aspects of step 4—the setup and maintenance of the software. ▪ Managers and procurers would also typically lead the project management aspects of the effort, including setting the overall software adoption and procurement timeline, establishing the tasks/activities to be completed during the timeline, pinpointing the responsible parties for each task/activity, and ensuring the entire effort stays on track.
Users	▪ User groups will provide input into the required and optional features and functions of the software, covered in depth in step 3. The software adoption process lead would oversee the process of documenting the features and functions with the user groups. They may also have input into identifying viable product options as step 3 makes progress. ▪ Topics to discuss with user groups will often pertain to the challenges and pain points they face as related to the software type in question. Understanding their challenges will lead to considering certain features and functions to help alleviate their issues.
Influencers	▪ For influencers, the topics to be discussed with them will be unique to the reasons why they are involved in the adoption effort.

Step 2c. Create a Tailored Information-gathering Process to Integrate Stakeholder Findings

Gathering information from stakeholders, overseen by the software adoption process lead, would typically be handled during virtual/in-person meetings and events. Planned meetings and events should be built into the adoption process and procurement timeline to ensure the input gained aligns with the decision-making process. Communication methods should be tailored to each stakeholder group and individual stakeholder as much as possible. Points to consider for approaching this process are shown in the table on the following page.

Stakeholder Category	Points to Consider
Managers and Procurers	▪ Managers and procurers would likely meet informally during meetings to discuss the software scope as covered in step 1. A series of meetings over a period of a few months might be planned, for example, to cover the step 1 items one at a time with space between the meetings for research to answer questions that arose. ▪ As the effort develops, they would meet to draft and refine the overall software adoption process timeline, the procurement timeline aligned within it. ▪ Users and influencers could be involved in some of these meetings, if their input is needed on certain topics. ▪ Stakeholder findings, the results of these meetings, would be integrated into project documents such as a draft resource summary and the software adoption process timeline.
Users	▪ In order to gain input on the required and optional features and functions of the software, the software adoption process lead would plan collaborative events. Rather than have all user groups present at one big event, it may be more effective to have a smaller event for each user group. This way, the content can stay very focused on what each user group needs to discuss. ▪ In the trip booking and scheduling example above, there were two staff member user groups mentioned: vehicle operators and service operations administrators. Members of the public, as app users and customers, were also identified as a user group. ▪ One way to structure the interactions would be to hold a series of small collaborative events, perhaps in two rounds for each user group. The first round would involve the software adoption process lead explaining the effort, the user group's role within it, and an open conversation about which features and functions they'd like to have included. ▪ The software adoption process lead would work between the two rounds to distill the information into key points, which would then be summarized at the beginning of the second round of events. ▪ The user groups would have an opportunity to confirm the direction, and add on details as needed. The second round could close with a discussion about the anticipated next steps for each user group, such as their involvement on the selection committee during the procurement process. ▪ Stakeholder findings, the results of these events, would be integrated into the required and optional features and functions document, a prominent driver in the procurement and software selection process. ▪ Contact would be kept with the user groups until they become users of the new software, and afterwards to get their input over time on changes needed.
Influencers	▪ Depending on the reasons why they are involved in the adoption effort, influencers may take part in the meetings and events mentioned above. ▪ If the topic to which they are contributing is separate from the topics involving managers, procurers, and users, separate meetings or events could be planned for them.

Aligning user group needs with specific software features and functions can be a challenging process, especially when the user group includes members of the public, but also for internal staff in some cases. If this becomes too difficult to handle internally, an agency may benefit from hiring a consultant specialized in user experience. Such professionals formalize the connections between the needs that users have for software and what a software can provide to help make the most effective match between the two—a promising investment for particularly complex software adoption efforts.

It can also be challenging to gain access to members of the public in order to gain input. An agency may have a list of members of the public they commonly reach out to who serve as representative stakeholders on certain topics such as service changes or policy changes. Such a list could be leveraged for software topics as well; it could potentially be located by inquiring with staff members across departments to find out if one exists. The software adoption process lead could also check into local organizations with regular events where such contact could be made. In addition, partner organizations may have advisory panels or other organized groups that include such contacts.

Working with stakeholders is all about relationships. The software adoption process lead will be the primary person responsible for managing the wide range of relationships involved in working with stakeholders. Early on in the software adoption process, these relationships will be established, but as the process goes forward, they will continue building and further develop. It is important that the agency commit to incorporating stakeholder feedback into software decision-making processes, so that the stakeholders know their input was impactful.

Key Takeaways

- Create a stakeholder map by listing the managers and procurers, users, and influencers of the new software, including the organizations and known roles/individuals at the organizations. Hold brainstorming sessions with colleagues or partners to fill the gaps if some contacts are missing initially.
- Identify key topics for each stakeholder group by reviewing Steps 1, 3, and 4 and pinpointing which topics would be of most importance to each group.
- Create a tailored information-gathering process, and integrate stakeholder findings back into Steps 1, 3, and 4 to close the loop between information gathering and decision-making.

Illustrative Project

GWINNETT COUNTY TRANSIT

Gwinnett County Transit (GCT) is the transit agency serving Gwinnett County, Georgia, part of the Atlanta metro area with an estimated population of 936,250.[12] In 2018, GCT decided to pursue a microtransit pilot project in Snellville, a municipality of around 20,077 people in Gwinnett County. The microtransit service was planned to serve the general public, without any eligibility restrictions, and GCT also had plans to expand into at least one more microtransit service area in the coming years. The idea of having a pilot project, prior to a wider service rollout, was viewed as a logical way to approach a new service type in Gwinnett County—allowing time to work out issues early in the operation. GCT was experienced in providing demand-responsive ADA paratransit service, but starting on-demand microtransit that used advanced software would certainly involve a learning curve.

GCT put in place a series of activities to gain input from key user groups, especially the GCT staff members who would be using the platform daily. Vehicle operators and their supervisors were one key group, managers and administrators of the overall service were another, and customer representatives were included as well. GCT tailored an information-gathering process to ensure the points of view of these groups were incorporated into decisions.

The pilot project ran for 8 months, from September 2018 through April 2019.[13] Having the pilot project for 8 months, prior to expanding the service into a wider program, provided GCT with a unique opportunity. The agency could glean information based on real day-to-day operational experience prior to embarking on a procurement process that would commit the agency to a software platform for a year or more. During the third month of pilot project operation, GCT held informal group interviews with the vehicle operators and their supervisors. Since they had already been providing trips for a few months, they had a lot to say about what was going well and not so well, as well as reflect on typical customers and share their general observations.

For example, the vehicle operators mentioned that they wanted the ability to see in advance if an upcoming passenger would require assistance with a mobility device, which was not possible in the pilot platform. This information helped them to mentally prepare for what was coming next. They also wanted the option to provide information when they needed to stop for unplanned maintenance needs or other purposes, so that the reason, precise timing, and location of the unplanned stop could be well documented. Further, the pilot software did not track all the details of the routes, such as the period between departing from the transit headquarters and officially starting the microtransit route, and this functionality was requested. Since asking about the gender, age, and other details of the passengers was deemed too personal to request voluntarily via the public-facing app, the vehicle operators shared insightful stories about the passengers. Passengers often spoke with the operators and told them what they thought about the new service. The operators shared that a number of passengers mentioned appreciating the independence the service afforded them. There were visually impaired passengers who typically did not leave home as often without the service, and older passengers who were going to the movies and other places more often just for fun, for instance.

Managers and administrators of the service had additional ideas about what the platform should

[12] United States Census Bureau. Quick Facts. Available at: https://www.census.gov/quickfacts/ as of February 10, 2021.
[13] Gwinnett County. Test program for microtransit bus service concluded April 30. Available at: https://www.gwinnettcounty.com/web/gwinnett/home/stories/viewstory?story=Testprogramformicrotransitbusserv as of February 10, 2021.

include. For example, they wanted the data reporting process to the National Transit Database (NTD) to be more seamless—transmitting the data points from the software to NTD directly with limited involvement on their part. They appreciated the wide range of reporting and data options the pilot platform had, but wanted a better fit for "executive decision-making" purposes, such as a daily dashboard of the top 5-10 indicators that needed to be continually monitored. They wanted more data on where trips were going around Snellville (i.e., daily/weekly/monthly "hotspots") as well as details on trips that connected with fixed-route options in the GCT service area—to better understand how the microtransit service supported cross-area connectivity. In addition, they wanted key data points to be compiled in a way to help "diagnose" potential reasons for no-show trips, when a passenger booked a trip but never came to take it.

Image source: Gwinnett County Transit

Stakeholder findings were integrated into the heart of the software platform selection process, the Request for Proposals (RFP) GCT planned to use to guide the procurement. Unfortunately, GCT was unable to scale up to a microtransit program after the pilot finished, although GCT's management considered the pilot highly successful and popular with passengers. Nonetheless, the microtransit software platform RFP document, which was drafted during the pilot period, was broken down into the typical software components along with key features and functions that the stakeholders requested—marked as either "required" or "encouraged." In this way, the information gathered from stakeholders was a direct input into the procurement and decision-making process. Further, GCT planned to have live demonstrations of the software options to verify the features and functions, also planning for the stakeholder groups to play a strong role during the demonstrations by considering more subjective aspects of the software such as ease of use. The various users of the software were put in the driving seat, so that they had a strong voice in which software platform would have been selected.

Chapter 3
Step 3: Move Forward with a Software Product

This chapter opens with a number of topics that an agency should become familiar with in order to move forward with a software product. This information will prepare you to take the steps provided at the end of the chapter. The topics are shown in the table below.

Available Software Product Types
Computing Platform Considerations
Software Product Purchasing Options
Software Functional Types for Small Transit Systems
Software Requirements, Functions, and Features
Software "Ownership" Requirements
Software Affordability
Software Application Procurement

Available Software Product Types

When a small transit system obtains software to assist it in managing its services and operations, there are three types of software products that it may wish to consider. These options are described below, including their relevance for small transit systems.

1. Commercial off-the-shelf software, often referred to by its acronym, COTS

COTS software is explicitly a "product" that has been developed by a technology (software) company which not only sells the software, but also provides a license (or a usage agreement) that gives the buyer the right to use the software. The license agreement also warranties that the product will work as intended. In addition, the technology company supports the product for its customers. This includes periodically updating the product, including fixing any defects that are discovered in the course of customers using the software.

For purpose of this discussion, referring to a software product as a COTS solution has no implication for how that product is delivered to customers. As will be discussed subsequently, the recent trend toward software being provided to customers via the Internet from a location in "the cloud", referred to as Software as a Service (SaaS), still involves COTS software.

COTS software is sold for a price that is specific to the company that has developed the software. Different COTS products that have similar basic functionality may be sold for significantly different prices. The price of a specific product will depend on the scope of its functionality, the importance of the software's functions to the service being delivered, and the market position and reputation of the technology company. COTS software that is delivered via the SaaS model will have a different pricing structure than such software that is sold as a licensed product to a customer.

Core and optional software modules in COTS products

It is common for COTS products to include multiple software "modules". In particular, there may be a "core" product which provides the essential functionality for a specific business need as well as optional "add-on" modules which provide further useful functionality, but which may not be necessary for all customers. There is usually an additional cost to purchase each optional module.

Examples of COTS Products

DRT software applications provided by companies such as Trapeze Software, Routematch, Ecolane, CTS and others, are used by hundreds of transit systems to manage general public DRT or ADA paratransit services. Such COTS products have been on the market for many years, from the early 1990's in the case of Trapeze. A new generation of COTS products for DRT from recently established software providers has appeared during the past few years.

For some transit systems, the optional modules may encompass functionality that is a core element of how *they* deliver service, and hence are not truly optional for them. For example, a demand responsive service that provides customers with advance notification of vehicle arrivals—by email, text message, or automated phone call—will require such functionality in its software system, and if this is not included in the core product, it will be essential for the agency to purchase an add-on module for this purpose.

2. Open source/public domain software

Wikipedia defines open source software as "a type of computer software in which source code is released under a license in which the copyright holder grants users the rights to use, study, change, and distribute the software to anyone and for any purpose". *Source code* means the actual software programs that make up a software product.

As this definition implies, open source software does not require payment to be able to obtain the software—it is "free". As will be explained below, such software products are free only in the sense that a person or organization acquiring the software does not need to pay the entity that created the software for the source code.

It bears noting that open source software may have certain limitations on its use which will be specified in the license provided to those who acquire it. In particular, open source software is often restricted from being used in commercial software products. That means that if an organization obtains open source software and then develops—or pays for the development—of a new software application that also uses the open source code, it is legally restricted from selling the new software. This should not be an issue for the typical small transit agency, but it is important to be aware of the legal restrictions that often accompany open source software.

Public domain software

Public domain software is a close cousin of open source software. Neither cost anything to acquire, but public domain software has absolutely no property rights, and hence no related rights such as copyright, trademark, or patent. Public domain software can be modified, distributed, and even sold by anyone without attribution. It can be used without restrictions.

At the same time, an entity selling public domain software as part (or all) of a software product does not possess any license rights to the public domain software itself, and the buyer can modify that specific software without restriction or additional payments to the seller. Of course, the buyer could obtain the actual public domain software for no cost if it wished. It is typically paying the organization that provides the software it is using because the software producer has embedded the public domain software in a larger product or is in some other way providing additional value to the buyer, such as configuring, supporting, and enhancing the software.

Why open source/public domain software is typically *not* costless

As indicated previously, open source software—and public domain software as well—can be obtained and used without paying the entity that developed the software. But that is the only sense in which such software is "free".

In contrast to COTS products that are purchased from the organization that has developed them and which are accompanied by a warranty that legally stipulates that they will work as intended, open source/public domain software is obtained "as is". The acquirer is totally responsible for installing and configuring the software, testing it with one's own data, fixing anything that does not work properly, and supporting it once it is being actively used. These are all significant technical activities requiring competent, experienced software professionals in order to be performed satisfactorily. Clearly there are costs associated with engaging such resources, and those costs could be (much) greater than purchasing a COTS solution.

It is *not* common for organizations to use open source/public domain software for their own business purposes *unless* they possess technology resources—whether in-house staff or technology company partners—that can work with the software. When open source software *is* used, the most common approach is to purchase a package of services from a software company that has demonstrated expertise with the software. Such companies implement the open source software for their clients, potentially modify and customize it for specific customer needs, and generally assist their clients in using it for their business purposes. These companies sell the ability for other organizations *to use the software without needing their own technical resources to be able to do so*.

Examples of Open Source/Public Domain Software

Open source and public domain software have to date been infrequently used by small transit services, but there are a few relevant examples of such software.

- **Ride Pilot—very basic DRT software, no automated scheduling**
- **1 ClickICS—open source mobility management software**
- **Denver Trip Exchange—public domain software that enables the DRT software applications of multiple agencies applications to post their unscheduled trips to a central trip clearinghouse where other agencies can claim such trips for delivery by their DRT operations—all trip data is automatically transmitted between the DRT software applications and the trip exchange.**

When open source/public domain software makes business sense for small transit agencies

Small city/rural/tribal transit agencies require software for specific purposes, and for most of those purposes there is a reasonable set of COTS solutions that can be purchased from technology companies with substantial experience in this sector of public transportation. Moreover, because this sector is relatively mature, the technology vendors' COTS products are priced competitively for the most part.

Obtaining open source/public domain software for core operations purposes in such circumstances is in *most* cases *unlikely* to be a sound business approach. There are certain situations, however, in which open source/public domain software may be relevant and appropriate, notably:

- Open source software that is provided by an established technology vendor which also configures, supports, and further develops the software as part of the product offering, for prices competitive with comparable COTS products.
- Coordination and/or integration of multiple agency's services, including mobility management initiatives.
- Transit services that are very basic and do not require COTS products that include more comprehensive functionality.

3. Custom developed software

Custom developed software is appropriate when the functional and business needs of an organization are not well met by any existing software product. While this is not uncommon for business situations, which can be highly diverse even within a single industry sector, it is *not* typical for a sector such as small city/rural/tribal transit where service approaches are very similar across organizations and there is a well-established eco-system of software companies with viable COTS products.

Where custom developed software *is* likely to be relevant is in situations where a basic product already exists but is not well-suited to the specifics of the transit agency's needs. The basic product could be either a COTS product or open source/public domain software. Using the existing software as the starting point for the custom developed solution could be a sensible business decision in such circumstances, as trying to fit a COTS software solution to a situation it is not designed for often leads to unsatisfactory results.

Determining whether custom developed software is a good solution for a transit agency

Determining whether a custom developed software solution is the most appropriate software for a small transit agency is a challenging exercise, as COTS software vendors will try to persuade prospective customers that their solution is adequate for their needs, even when it is only partially a good fit. It may be necessary for the agency to utilize external resources (e.g., knowledgeable staff members of state DOTs or larger planning or transit agencies) to help them assess their situation if they do not perceive that existing products are truly suitable for their situation.

Statewide or national organizations such as CTAA may also be of assistance, particularly regarding the possibility that an open source software application could be an appropriate starting point for a custom developed software system. These organizations may be more familiar with the open source software that is applicable, which is limited. The advantage

Examples of Custom Developed Software

The open source/public domain Ride Pilot and Trip Exchange software applications are examples of custom developed software. The Ride Connection Clearinghouse software that is the predecessor and partial model for the Trip Exchange system was custom developed.

of developing a custom software application using open source software is that it is likely to be less expensive—perhaps much less expensive—than customizing an existing software product. But this is predicated upon the existing open source software already including much of the core functionality that the agency needs.

Small city/rural/tribal transit agencies find themselves in many different sets of circumstances, and should not assume that just because their peers use COTS software that there will be a product that meets their specific needs. Custom developed software—particularly a variant on an existing product—can be a much better choice than a poorly fitting COTS product. But custom developed software *will* cost more than a COTS product—if one exists—*and* require technically informed and substantial agency involvement in and oversight of the software development process. Technically knowledgeable external resources can augment the agency's own staff in managing the process, but this will involve additional costs. This is irrespective of whether the starting point is a COTS product or open source software—the latter will reduce the financial impact, but not the need for strong agency involvement and/or technology consultant oversight.

Only agencies willing to accept these realities should consider this option when procuring software. Custom development is only for the brave.

Computing Platform Considerations

Until about a decade ago, a small transit agency purchasing a software product also needed to provide the "computing platform"—the computer hardware, networking devices, and other underlying hardware and "system software"—which is used by the actual software application that the agency is procuring. The computing platform is an essential element of any software system.

Today, there are multiple ways in which the computing platform can be implemented, with important implications for the transit agency which is procuring a software solution. The 3 major options for the computing platform are:

1. On site hardware and networking equipment (typically referred to by the acronym CPE, which stands for Customer Premises Equipment)
2. Hosted by a third party—which can be arranged by either the customer or the software provider, most typically the latter in the case of small transit systems
3. Hosted "in the cloud" by a Software as a Services (SaaS) provider who provides the computing platform as an integral part of its software offering

Recent Trend Toward Hosted Approaches for Computing Platforms

The strong trend during the past several years has been toward software being hosted "in the cloud" rather than on the customer's premises. This does not mean that this is the most appropriate solution for every small transit agency, but there is a compelling reason for this trend. Namely, the CPE approach requires the agency to assume responsibility for the core hardware and local networking of computer workstations (personal computers used by staff members to access the software application). This is clearly technically burdensome for a small agency, as well as associated with performance risk. In contrast, hosted computing platforms move the responsibility to a third party and the agency simply needs to understand how to use the software application and ensure that users have reliable Internet access—the remainder of the technology infrastructure is not their concern.

Potential Disadvantages of Hosted Approaches

Hosted approaches work very well when an agency has reliable high speed access to the Internet, i.e., access at broadband speeds. Broadband typically uses technologies such as cable or DSL which have large data transmission rates; the FCC defines

broadband transmission rates as 25 million bits per second for uploading data and 3 million bits per second for downloading data.

The most recent FCC Broadband Progress report finds that approximately 19 million persons—6 percent of the population—still lack access to broadband service. In rural areas, nearly one-fourth of the population —14.5 million people—lack access to this service and in tribal areas, nearly one-third of the population lacks access. Unfortunately, these data do not enable one to determine the situation in small communities—as opposed to the rural areas outside of those communities—since small transit agencies (tribal systems are a somewhat different matter) are almost always located in an actual city, albeit in many cases a very small one.

If an agency does not have reliable access to the Internet at near-broadband speeds at a minimum, it should probably not consider a software product that can only be delivered via a hosted solution. In such cases the agency will most likely be best off with an on-site computing platform. It is increasingly the case, however, that even small cities in rural areas have broadband access.

Software Product Purchasing Options

As a result of changing business practices in the software industry, as well as the transformative impact of cloud computing, software products are today sold via one of 3 different models, which are explained below:

1. Licensed software product (COTS) with annual support and maintenance fees.
2. Externally hosted version of the licensed software product—the software company (or a third party it designates) is responsible for the hosting arrangements.
3. Software as a Service (SaaS).

1. Licensed software products

Until recently, COTS software products sold to small transit agencies were almost exclusively licensed software products. With a licensed COTS product, an agency is provided with a copy of the software that is installed on hardware in its own computing environment—typically the CPE approach, although essentially the same situation for third party hosting—and the agency receives a license to use that software for some defined period, perhaps indefinitely (usually referred to as a perpetual license).

Perpetual software licenses provide the agency with the ability to use the software for an indefinite period of time, but usually only the specific version that the agency purchases. Moreover, purchasing the license does not provide the agency with access to support for the software nor any maintenance of the software beyond the warranty period, which is a defined period of time specified in the license agreement. The warranty promises that any defects in the software that are discovered will be fixed at no cost to the agency. Consequently, an agency will typically pay the additional fees for support and maintenance, since it wants to have a defect-free, reasonably current version of the software and the ability to communicate with the software vendor's technical support team if it experiences difficulties in working with the software.

Each software company has its own practices—which are spelled out in the license and support agreement that the agency signs—about upgrades to licensed COTS products. Sometimes the support and maintenance contract enables the agency to obtain new upgrades at no additional cost; many software companies require additional payment for major upgrades.

2. Externally hosted licensed software

An externally hosted licensed software product will have essentially the same licensing terms. The only difference is that the contract will include payments for hosting of the software in an environment which the software provider is responsible for, and there will be certain guarantees of performance for the computing platform. These guarantees of computing platform performance will typically be in the form of a Service Level Agreement (SLA).

An SLA will include such criteria as "up-time" guarantees, e.g., 99.9% uptime. This means that the software system will be operational 999 out of every 1000 hours—or only 1 hour of "down-time" (when the system is unable to function) in 71 days if the service operates 14 hours per day. SLAs may also include criteria for response times for users and the average and maximum times for scheduling customer bookings for a demand responsive service. One of the major advantages of using a hosted version of the software are these SLAs, which provides the agency with a contractual assurance that the reliability and performance of the software will be adequate for its needs.

3. Software as a Service (SaaS) arrangement

In a SaaS arrangement, the agency does not actually acquire a licensed version of the software, as in the other 2 models. Instead, it pays for the right to use a hosted version of the software for a specific period of time, typically a minimum of 12 months. The actual software itself will usually be essentially the same as a licensed product. In essence, the agency is "renting" the ability to use the software for the duration of the SaaS arrangement.

The advantages for the agency in this model are two-fold. First, the computing platform is provided by a third party via an arrangement with the software vendor. Typically, this will be a cloud-based platform provided by large organizations such as Amazon Web Services or Google Cloud which provide high quality, very reliable computing environments. Second, the agency will automatically receive updates and upgrades to the software whenever the software company improves its product, which tends to be relatively frequent for typical SaaS arrangements. There is no support and maintenance agreement necessary, as all customers use the same, most recent version of the software.

While all SaaS customers of a specific software application use the same software, each has the software configured to their specific needs and circumstances in the cloud-hosted computing environment. Each customer typically has a separate instance of the software application operating in the cloud environment, although sometimes certain computing infrastructure will be shared even as each customer's data is maintained separately.

SaaS vs. Traditional Models of Software Ownership: What to Consider

The SaaS approach has transformed how software is provided to customers in all industries over the past several years. Most contemporary software applications are now designed to be delivered via the SaaS model. For small transit agencies with broadband Internet access, there are few if any significant non-financial disadvantages to the SaaS model. Moreover, a SaaS approach provides at least 3 significant advantages to an agency.

1. The agency can change software vendors—and applications—with minimal complications if it is not satisfied with the performance of the software that it is using.
2. The agency avoids the direct equipment (hardware) costs and staff/technical responsibilities required to host the software.
3. The agency gets software support and automatic upgrades without doing anything.

At the same time, it is important for small transit agencies to recognize that the "lifetime" costs of software purchased via the SaaS approach *may* be somewhat greater than the licensed product approach, as illustrated by the following hypothetical—albeit realistic—comparison.

- A SaaS application that costs $1500 per month plus a $6,000 initial setup and configuration fee will cost an agency $132,000 over a 7-year period.
- The same software application purchased for a $40,000 license fee and an annual support/maintenance fee of 23% (the approximate norm for software) of the purchase price, plus one major upgrade of the software for an additional $15,000, will have a 7-year cost of $119,400.
- Over a 7-year period, the licensed version of the software costs approximately $12,500 less than the SaaS approach, a cost advantage of approximately $1800 per year.

Image source: EZ Ride located in New Jersey

- If the likely lifetime of the software is more than 7 years, the cost advantage of the licensed product would be greater.
- However, there would be additional costs for the computing platform for the licensed product approach, as well as more technical responsibilities for the agency, and this should be considered in any financial comparison with the SaaS approach. It is likely that these factors would substantially outweigh the modestly lower annual costs in this hypothetical situation.

Software Functional Types for Small Transit Systems

Small city/rural/tribal transit systems provide different types of transportation services, but all include 5 common elements: (1) vehicles; (2) drivers; (3) customers; (4) a specific service modality (e.g., fixed route service, demand responsive service) is used to organize the operation of the service, with the understanding that small agencies frequently provide both fixed route and demand responsive service; (5) data is generated by the operation of the service, and the performance of the service can be understood by means of the analysis of this data and reports produced from such analysis.

The software needed by a small transit agency will ideally encompass each of the above areas. It is not necessary for a single product to include all of the important functionality. For example, vehicle maintenance software applicable to many different types of vehicle fleets is widely available and can be obtained separately from software that is specific to an agency's service delivery system.

Two other types of software have become increasingly important during the past decade, and for some agencies they may also be considered as essential for their core needs.

(1) Trip planning software: As consumer use of Web-enabled map-based tools—including smartphone apps—for determining how to get from point A to point B has become commonplace, trip makers interested in using transit increasingly want to be able to plan their transit trips in advance, including trips that involve multiple modes of service.

(2) Digital fare payment systems: As consumers have increasingly turned to non-cash forms of payment, using either cards or mobile phones, for their everyday needs, they would prefer to do this for public transit as well. Hence software for fare payments has become an increasingly important element of both fixed route and demand responsive services.

It bears emphasizing that for very small agencies, e.g., 10 or fewer vehicles, not all essential functions must be included in the core software that they purchase. It is often the case that an Excel spreadsheet application can be used for basic purposes such as record keeping for drivers (e.g., recording their driver's license numbers, special operating certificates, training courses completed, etc.). Even weekly driver schedules can be maintained in Excel for very small systems with little disadvantage. There *can* be advantages of having all of the necessary functionality in a single software system, but the more functionality included in a software product, the greater is the cost typically. Small agencies with limited budgets need to be pragmatic about what functionality is necessary for their core software system.

The most important software needed by a small transit agency is that which enables it to manage the transportation operations which deliver its core service(s). There is an important distinction between fixed route services and demand responsive services in this regard; the key functional software needs for each of these service modalities are summarized below.

Software for Fixed Route Transit Operations

A fundamental need of small transit agencies which operate fixed route services is for software that monitors real-time vehicle operations. A common name for this software is CAD/AVL—the acronym stands for Computer Assisted Dispatch/Automatic Vehicle Location. Such software, which requires a GPS-enabled device in each vehicle with wireless connectivity that enables it to transmit its location frequently (e.g., every 15 seconds), is able to track the location of all vehicles in service and make this information available—via a user-interface, which may be map-based—to both the operations staff of the agency and the public. For the operations staff, the software application informs them whether the vehicle is adhering to its schedule or not; it may also make predictions of when the vehicle will arrive at downstream locations on its route.

There is likely to also be a customer facing application—web-based—associated with CAD/AVL software that informs the (prospective) transit

Image source: Pelivan Transit located in Big Cabin, Oklahoma

user of the vehicle's current location and when the vehicle is both scheduled and predicted to arrive at the stops along its route. This application will typically show the vehicle's location on a map. The same software vendor may sell both the CAD/AVL system and the customer-facing application, or the CAD/AVL vendor may partner with another company that has developed the latter software. It is possible that such a software application is also available in a smartphone app-based version.

Small transit agencies may wish to purchase both the operational software and the customer facing software together. If they are purchased separately, the CAD/AVL system should be purchased initially, and then the customer facing application can be procured, with one of the requirements being that the software selected must be able to be integrate the locational data feed.

Other fixed route transit software that may be needed by a small agency will be that which generates GTFS data for its operations—including its schedules—and creates a data package that can be utilized by Google Maps and other third party applications for trip planning purposes. Such capabilities can also be purchased as a service from a few technology companies. The availability of transit trip planning can be very valuable for local residents and visitors who want to know what their transit options are for a specific trip and how to travel from point A to point B on the bus.

Small fixed route operators may also need some type of software for driver management (including generating driver schedules) and vehicle maintenance, but as noted previously this could be as simple as an Excel application developed by a knowledgeable staff member. Both driver management and vehicle maintenance software are also relevant for DRT services, and are discussed following the section below on DRT software.

Software for Demand Responsive Transit (DRT) Services

Small transit agencies that provide DRT services require a more comprehensive software application than for fixed route operations. This is because a DRT operation includes customer booking of trips, scheduling and dispatching of vehicles with at least some real-time elements, and increasingly an application used by the driver that directs his/her activities. In addition, the DRT software application will include an administrative component that encompasses service configuration settings, driver schedules, and vehicle management.

During the past few years the functionality and sophistication of DRT software has improved substantially, in important part due to the entrance of a number of recently established technology companies into this market with more advanced products. This has stimulated competitive responses by the established software companies. Many DRT software packages now include self-service trip booking by customers via a web-based application or smartphone-based app. Fully automated scheduling and minimization of manual dispatching is another hallmark of the newer software applications.

A contemporary DRT software system will include the following functionality:

- Customer information and registration (can be self-driven)
- Trip booking—by agents at a minimum, and self-service if appropriate
- Trip scheduling, consisting of the automated scheduling of passenger trip requests onto specific vehicles with estimated arrival times at each pickup and drop-off location, and the routing of vehicles from one stop location to the next.
- Vehicle dispatching, including the transmission to the driver application (explained next) of each pickup and drop-off location in driving sequence as well as the identity of the individual(s) being processed at each such location.
- Driver application, which encompasses driver manifest management, display of routing information (usually map-based) to help the driver navigate from stop to stop, recording

of passenger arrivals and departures on the vehicle, and registering of fares collected including the type of fare.

- Administrative module, which includes functionality for the agency staff to be able to manage the service configuration and day to day operations:
    - Service configuration—service zone boundaries, operating hours, type of DRT operation (specific address to specific address, fixed and virtual checkpoints, sub-zones, locations visited on a scheduled basis, etc.)
    - Operational schedules—starting times and ending times for the service zone by day of week, exclusion of days as appropriate
    - Driver management—driver schedules and information about drivers' qualifications for different types of vehicles if relevant to the operation
    - Vehicle management—including vehicle deployment schedules, vehicle capacity, and vehicle characteristics (such as wheelchair lifts)
    - Data management and data reporting, including management reports

Software for Vehicle Fleet Maintenance/Asset Management

An agency's vehicles are its core physical asset—without vehicles, there is no transportation service. Vehicles have certain key characteristics—manufacturer, model year, engine type (gasoline, diesel, electric), number of seats, equipment to handle wheelchairs, etc.—and a finite life span that is primarily dependent on how much they are driven. Vehicles require periodic maintenance to renew and replace parts and fluids they use.

Vehicle fleet maintenance software enables a transportation operator to record and track information on each of the vehicles in its fleet, including the specific equipment that is installed on each vehicle and when maintenance and other key events—such as renewing vehicle registration—are scheduled to occur. Software can be quite detailed, including functionality to record when certain parts are replaced, manufacturer and part number of the replacement part, and how much it cost.

Agencies have 3 options for vehicle fleet maintenance/asset management software: (1) purchase a stand alone fleet maintenance product (SaaS options are now available); (2) purchase a fleet maintenance module that is included with a fixed route or DRT software product; (3) use a tool such as Excel to manage the basic fleet maintenance data for its vehicles. There is a very robust ecosystem of software companies that provide fleet maintenance software, some of which is very appropriate for the needs of small transit operators and is not expensive. At the same time, if a DRT or fixed route product that is of interest to the agency has a fleet maintenance module that is competitively priced with the stand alone fleet maintenance software option, there can be advantages to using a single product.

Software for Driver Management

Driver management software encompasses functionality for both recording and displaying information on drivers and their key attributes, most notably their training and qualifications for driving certain types of vehicles and handling different types of clients (riders), and driver scheduling. It is not uncommon for basic driver management functionality to be included with DRT and fixed route

software, although this does not usually include the ability to automatically generate optimized weekly driver schedules meeting specified criteria. Many small agencies manage their driver schedules and information about their drivers in Excel. As with vehicle maintenance software, there are advantages to having driver management integrated into the core software product used by an agency but it is not essential. Using Excel for such purposes is a reasonable alternative for agencies with fewer than 20-25 drivers.

Software for Fare Payments including Mobile Payments and Ticketing

In the past few years, several software companies have developed fare payment systems that enable passengers to pay public transit fares via mobile phones. As smartphone penetration now approaches 85% of the adult population, it is clear that most transit users are able to pay for a transit ride via a mobile device if this option is available. In addition, some mobile ticketing approaches merely require a basic mobile phone, not a smartphone.

The subject of mobile payments and ticketing is sufficiently broad and complex that it is beyond the scope of this Guidebook. Nonetheless, it is important for small transit agencies to carefully assess their fare payment situation when they are considering acquiring new software—if they have not done so as a separate priority. Particularly if they are acquiring new DRT software, there are mobile payment solutions that are readily integrated with such software and it may make sense for an agency to move to a new fare payment system at the same time.

Software for Trip Planning

Another major recent development in software for transit services has been the advent of trip planning software, often based on a software platform named Open Trip Planner (OTP). OTP itself is open source software, and commercial products have extended its functionality. The purpose of trip planning software is to provide the trip maker with the ability to use their computer and/or smartphone to pre-plan a trip that involves using public transit. Google Maps provides basic transit trip planning capabilities for fixed route transit if a transit agency has made its GTFS data available. Trip planning applications that are optimized specifically for public transit include more extensive and in-depth capabilities. Examples include trip planners from companies such as Transit (Transit App) and Moovit, which are now available to the public at no cost in many larger cities.

Small transit agencies do not necessarily need their own software to take advantage of trip planning functionality. With the appropriate GTFS data provided by the local agency, Google Maps will automatically include public transit options in the trip plans it generates for its users, even in small cities.

At the same time, if a small transit agency wants to tailor trip planning for its own purposes, it will need to identify a commercial or open source software application that it can use. Particularly for agencies which operate both fixed route and DRT services, trip planning software offers the potential of making services more easily discoverable by local residents with information more tailored to their specific trips. Moreover, OTP-based trip planning software is now able to make trip makers aware of the presence of DRT services via the GTFS-Flex extension of the GTFS data standard. This makes possible the integration of trips plans with DRT software, creating the ability for a customer to plan a trip and then—if the trip involves using DRT in any way—book the DRT component via the same user application (assuming the DRT software is in fact integrated with the trip planner software).

For most agencies, orchestrating the implementation of an integration of trip planning software with its DRT or consumer-facing fixed route software is likely to be beyond their capabilities, but the software companies themselves are moving in this direction. Even before the end of 2021, it is possible that such integrated software products will be available to small transit agencies. As they do become more widely available, they will represent another type of core software for small agencies.

Inter-Operable Software Considerations: A Short Discourse

Some small transit systems not only provide transit service but also function as Mobility Managers for a variety of transportation services—not just their own—in their service area. Such services can include those provided by human service transportation (HST) organizations, those funded by Medicaid (and delivered by service providers via Medicaid transportation brokers), and various other local services, including those relying on volunteer drivers.

In order for Mobility Management programs to be able to function as efficiently and cost-effectively as

possible, the software systems used to manage the different transportation services—with are mostly or exclusively demand responsive in nature—need to be able to inter-operate, at least at a basic level. Currently, this is only rarely the situation—inter-operability is a desired state, not an actuality.

As discussed in Chapter 1, inter-operability between software systems requires other software—typically in the form of Application Programming Interfaces (APIs)—that enables them to not only share data, but to do so for specific functional purposes. For example, one DRT system—call it System A—may wish to transmit a trip order to another DRT system—call in System B—and if the latter system has an API it can do so in a standard way. However, System A must communicate with System B using the language of the latter's API—call it API B—and is restricted to using the functionality made available via API B. This will require new software development for System A. Even if System A has its own API—call it API A—that will not be sufficient for it to inter-operate with API B, as API A has its own functionality and language syntax which is highly unlikely to be the same as API B. The situation is shown below.

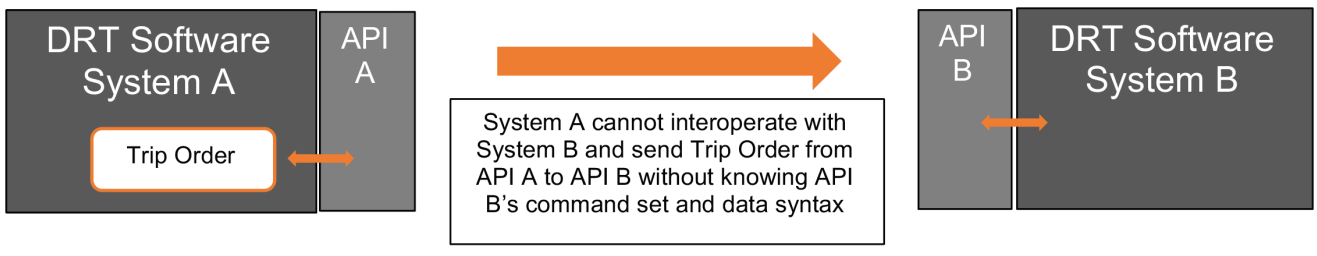
The situation described is actually a best case scenario in the sense that both software systems already have the capability to inter-operate with other systems. But actual two-way communication between System A and System B can only occur after both have developed additional software capabilities to use the command set and data syntax and data requirements of the other's API—only then will it be possible to send trip orders and data messages back and forth. Moreover, if System A wants to perform some functional action which System B's API does not support—such as updating a previously transmitted trip—then yet more software development will be needed by one or both parties to make such functionality feasible.

As this discussion illustrates, achieving inter-operability between software systems is not a simple matter, even when all parties are committed to this outcome. Resources and time will be required if the relevant software systems are not already inter-operable with each other. The current reality is that there is very little "off the shelf" inter-operability among the software systems used by small transit agencies, particularly for DRT systems. For fixed route software, inter-operability is less complex, such as a map-based interface to show the location of vehicles on routes and their schedule adherence—where the data is sourced from the CAD/AVL software of one company even as the map-based interface is the software of another company.

Given that inter-operability *is* a very important objective for some small agencies—particularly those providing DRT services—who participate in Mobility Management systems, what can such agencies do when they are purchasing software to advance their systems towards this objective? One answer is to select a product from a software company that has already made clear by its actions that inter-operability is important.

In particular, if the software product has an API—even though that does not ensure inter-operability with another software vendor's product—it is evidence that inter-operability is a priority. It is also an indicator that the company uses contemporary software approaches, which is a pre-requisite for technically efficient inter-operability with other software platforms. In addition, if a software company is able to document that it has actually achieved some level of data integration with the software product of another company, that is further evidence of its commitment to inter-operable software.

In view of the currently fragmented nature of software solutions for small transit systems, and the absence of any data standards to facilitate inter-operability among such software, inter-operable software is likely to remain an aspiration rather than a reality for at least the next 12 to 24 months.



Software Requirements, Functions, and Features

Before a small transit agency initiates the process of searching for, and eventually acquiring, a new software system, it must determine the essential capabilities of the software. The prior section has briefly described the different types of software applications that are relevant for small agencies. Knowing what type of software it needs, the agency then must determine what its core requirements are for that specific type of software.

The IEEE Standard Glossary of Software Engineering Terminology defines a *requirement* as: *A condition or capability needed by a user to solve a problem or achieve an objective.*

- *Features* and *functions* of the software are the means by which requirements are met.
- *Features* are the "tools" you use within a system to complete a set of tasks or actions.
- *Functionality* is how those features actually work to provide you with a desired outcome.
- *Functional requirements* define the basic software system behavior. Essentially, they are what the system does or must not do. An example is shown below.

It is the agency's task to document each of the important functional requirements for the software they intend to acquire. In addition, the agency should identify and document all features which are important to it. For example, if the software should be able to show the current location of all vehicles in service on a map, and for each vehicle provide a list of passengers currently on board, this feature must be specified as required by the agency.

All of the important requirements, features, and functions must be documented and included in the RFP document. Collectively they represent the target for the software that the agency intends to acquire.

Software "Ownership" Requirements

As every organization which uses software to assist in the performance of its core functions is aware, the activities associated with acquiring, implementing and using such software impose significant resource commitments over and above the direct cost of the software application. Small transit agencies, who are typically resource constrained, need to carefully assess their situation vis-à-vis these software ownership requirements as they embark on the process of acquiring new software to manage their services and operations.

It bears emphasizing that the activities and resource commitments discussed below occur *after* the decision has been made to acquire a specific software application. However, these factors should be considered *prior* to initiating the software acquisition process.

In particular, the agency needs to consider how these factors are likely to be impacted by the different software products under consideration—the impacts are *unlikely* to be identical. In particular, there is a significant difference between a SaaS product and a licensed software product for some of these factors. Achieving clarity about the

FUNCTIONAL REQUIREMENTS

EXAMPLE: SCHEDULE A TRIP

- Requirement: Assign trip to a specific vehicle which is compatible with the passenger's physical mobility capabilities
- Scheduled time on board the assigned vehicle cannot be more than 2.5 times the direct driving time from passenger's origin to their destination
- Trip assignment cannot move pickup and/or delivery times of other passenger trips previously assigned to the vehicle outside of the time windows that have been originally communicated to the passengers
- If passenger's requested pickup or delivery time cannot be met due to no capacity being available, the passenger shall be offered the nearest time to their request
- System will provide passenger with pickup time window that extends 15 minutes beyond and 5 minutes prior to scheduled pickup time

agency's capabilities and situation will be important in determining which of those options is more appropriate.

The following major activities/resource commitments are key elements of software "ownership" after an agency has purchased a software application:

- Software system implementation—including the computing infrastructure if necessary.
- Software system configuration
- Staff training—both internal staff and any affiliated organizations.
- Staff time necessary to learn how to use the software at a high level of competency, including using it to monitor and improve operational performance.
- Handling new versions and upgrades of the software.

Software System Implementation

New software systems don't just happen. An agency acquires a new software application and then must implement it for its own purposes. If it has decided to use a SaaS software solution, implementation will be simpler and less burdensome than if it has purchased a licensed software product that will be installed on its own computing infrastructure.

In the latter case, the agency must determine its specific hardware and networking requirements, obtain and install the hardware (unless they already own the necessary hardware), install the actual software, and then ensure that the infrastructure works properly by testing the system. Technical resources are needed for at least some of these activities. The software vendor may be able to provide some of these resources; local information technology firms are another possible source of such resources, and some agencies may have staff with competency for at least some of the necessary tasks.

In some cases, software vendors will arrange for the hosting of the system and this option is comparable to a SaaS approach in terms of relieving the agency of the responsibility for the details of implementation. For agencies which do not have easy access to the resources needed to implement the necessary computing infrastructure, the SaaS option or third party hosting of a licensed product are likely to result in a better outcome.

Software System Configuration

Software applications must be configured and "localized" to an agency's specific circumstances. For a demand responsive service, this includes such factors as defining the service area boundaries and operating hours, setting scheduling parameters, determining the level of service (e.g., maximum wait time and ride time) for passengers, and setting up existing customers in the system. For fixed route services, data on routes and stops and timetables must be entered in the software, as well as driver schedules and information on vehicles. Since these data are known only to the agency, its staff will need to be heavily involved in implementation even in cases where the software vendor plays a significant role in system configuration.

Staff Training

Training the agency's staff in how to use a new software product usually begins a few weeks (sometimes as few as 1 to 2 weeks) prior to the agency actually using the software for production operations. Most software companies provide competent training to their customers, and a vendor-taught training class will almost always be part of the package of services that are included in a software purchase (as discussed in "Topic 8: Software Application Procurement").

If other organizations in addition to the agency need to use the software for certain purposes—such as entering trip requests or looking up scheduled rides—they will also need to be trained in how to use their specific functions. They can be included in the vendor-taught training class, but the agency will ultimately be responsible for ensuring that external organizations can properly use software functions that they have been given access to.

If the software system is something totally new to the agency—as opposed to replacing an older software package with similar functionality—there is likely to be a significant learning curve for the staff members who are using the software. In such situations, the vendor-led training will only be the beginning of the process of the agency learning how to use the software to its—and its customers'—best advantage. The level of effort needed for the staff to handle the basic features of a new software system is easily under-estimated.

Achieving Staff Competency in Use of the Software

As implied by the above discussion, it can be challenging for an agency's staff to achieve true competency in the use of a software application. But if the agency only has basic competency, many useful capabilities of the software may not

be able to be used, and the agency is not able to take full advantage of its financial investment. This is particularly the case with different configuration options in the software which, through proper use, may enable the agency to achieve improved operational efficiencies or performance. In addition, data management and analysis made possible by the software may be able to help the agency understand how to operate their services more efficiently and effectively, but the software system must be well-understood by the staff for such outcomes to even be possible. Agencies often do not take full advantage of the capabilities of their software, and intelligent software ownership will avoid such outcomes.

Handling New Versions and Upgrades of the Software

It is important that the agency be able to quickly move to a new version of the software when one is released by the software provider. A new version will include defect fixes as well or new or modified features and functions that have been included in the software in response to the requests of customers. If the software is a SaaS product, the customer will typically have no choice about using a new version, although the changes in SaaS products are typically incremental; licensed software products are more likely to have major new versions. In either case, an important part of the ownership of a software application is to be able to handle new versions of the software. The starting requirement for that is for the agency's staff to be highly competent with the current version.

Software Affordability

What does it mean when we say that a software system is "affordable"—or is not? This is an important question for every small transit agency contemplating buying new software or replacing the software it now uses with that purchased from another software provider. Small agencies are almost always resource constrained, and purchasing an expensive—relative to limited available funds—product immediately raises the question "can we really afford this software?"

The question of affordability has 2 dimensions. First, there will always be some practical upper limit on what a small agency can afford. It is often the case that small agencies obtain grants from state DOTs or other external funding sources to purchase software. The cost of the software system—including all related costs such as training, computing infrastructure/hosting, and implementation services—obviously cannot exceed the amount of the grant (if applicable) plus whatever surplus operating or capital funds the agency has available to allocate to the purchase.

Second, and perhaps more importantly, affordability should be thought of in terms of the value of the software relative to its "net out of pocket cost" to the agency. By net out of pocket cost is meant much more than the purchase price of the software. Net out of pocket cost is defined in the following table—with separate definitions for a licensed product and a SaaS product. In estimating the net out of pocket costs for the software purchase, the agency must also carefully consider whether there will be additional operational costs associated with acquiring

Cost Element	Licensed Product	SaaS Product	Notes
Initial Purchase Price	X		
Annual Support and Maintenance Fee	X		
Annual Software Service Fee		X	Multiply by expected life of product in years
Training Cost	X	X	
Installation of Software	X		
Implementation Services & Configuration	X	X	
Computing Infrastructure (Hardware & Networking)	X (?)		No cost if agency already owns necessary hardware
Application Hosting Fee (annualized)	X (?)		If hosted by vendor or third party, multiply by expected life of product in years
Major Product Upgrades (future cost)	X		Estimate number of major upgrades
Additional Agency Annual Operating Costs with Software (multiplied by years)	X	X	If expected due to changes in staffing or operations
Cost Savings Element			
Annual Operating Cost Savings	X	X	Multiplied by number of years of product life
Additional Fare Revenues	X	X	For estimated life of product

or using a specific software product—or conversely, a likely reduction in certain operating costs. Including such costs—or avoided costs—is essential in addressing the question of affordability.

For example, consider a new DRT software system that includes a web booking application and/or smartphone app that enables customers to reserve and schedule their trips without calling the agency, and which also includes sophisticated automated scheduling and dispatching that substantially reduces the need for a human dispatcher to manually schedule and dispatch trips. This combination of functionality might reduce the staffing requirements for the DRT operation by 0.5 FTE, resulting in significant cost savings. On the other hand, acquiring a software system that requires local computing infrastructure might require an agency to obtain technical services from a local information technology firm via a support contract. The cost of the service contract would be part of the net out of pocket cost of the new software.

After an agency has developed an estimate of its net out of pocket cost for the software, it is better able to evaluate the affordability of the software. Two examples are presented below (including the following page) for the acquisition of a DRT software system, one involving the purchase of a licensed software product and the other a SaaS product.

In both cases, the DRT application enables customers to make their own trip bookings, and includes fully automated scheduling and dispatching. These features make it possible for the agency to reduce its staffing for the DRT operation by 0.5 FTE, which equates to $18,000 in annual labor savings. Over a 5-year period, this totals $90,000 in reduced out of pocket costs.

As the two examples show, in both cases the net out of pocket cost of the software system is relatively low, which is attributable to the operating cost savings it makes possible. The two software options have relatively comparable annual net costs over a 5-year period, although the software acquisition costs used in this analysis are merely representative, and not indicative of what can be expected in any specific situation.

Is this software affordable? Given that its net cost per year is less than $7,000 in either case, one can make a strong argument that it is affordable.

Licensed Software Product Example

<u>Direct Costs of Software Acquisition:</u>	
Software Purchase Price:	$35,000
Annual support and maintenance Fee:	$8,000
Implementation, configuration, training:	$8,500
Vendor-Based Hosting fee:	$650 per month
Total First Year Cost of Software:	$59,300
Total 5 Year Cost of Software:	$122,500
<u>Other Out of Pocket Costs Associated with Software:</u>	
None	$0
<u>Out of Pocket Cost Savings Associated with Software:</u>	
Reduction in staff costs:	$18,000 per year
Total 5 Year Cost Savings:	$90,000
<u>Net Out of Pocket Cost of Software:</u>	
Total 5 Year Net Cost:	$32,500
Annualized Net Out of Pocket Cost:	$6,500

Another useful measure of affordability is the cost per annual passenger trip. If this DRT service was handling 120 trips per weekday (about 36,000 trips per year), the software cost would be no more than $.18 per trip irrespective of the type of the software. Even if there were no operating cost savings associated with the software, the net cost of the software system would be less than $.70 per trip. Given that the total operating cost for a small DRT service is likely to be at least $10-12 per trip, a technology related cost representing 6-7% or less of the operating cost would seem to be relatively affordable. A general rule of thumb is that the technology should not cost more than 10% of the total operating cost of a DRT service. For fixed route services, technology costs will typically be significantly less.

SaaS Product Example

Direct Costs of Software Acquisition:	
SaaS fee per year:	$21,600
Implementation, configuration, training:	$8,500
Total 5 Year Cost of Software:	$116,500
Other Out of Pocket Costs Associated with Software:	
	None
Out of Pocket Cost Savings Associated with Software:	
Reduction in staff costs:	$18,000 per year
Total 5 Year Cost Savings:	$90,000
Net Out of Pocket Cost of Software:	
Total 5 Year Net Cost:	$26,500
Annualized Net Out of Pocket Cost:	$5,300

Software Application Procurement

Transit agencies typically procure new or replacement software via a formalized procurement process. In most cases, as public organizations—or organizations using public funds—this procurement process must adhere to certain legal requirements which will be specified in state or local laws or regulations. It is not uncommon for certain legal requirements to only apply for purchases of more than a certain amount, such as $2,500. However, in virtually every case, there are laws and regulations that will affect how an agency conducts a procurement. In the following discussion, it is assumed that the software procurement will be conducted on the basis of formal proposals, including price proposals, from all organizations seeking to sell their software system to the transit agency. The actual publication of a request for proposals (RFP) is one of the final steps in the software procurement process.

Leveraging the base of information provided in the first part of Chapter 3, through the eight topics, consider what actions your agency can take to move forward with a software product. A series of seven steps is provided to walk you through the process, as shown below.

Step 3a. Determine What Type of Software Your Agency Needs

As discussed previously, there are several types of software which could be relevant to a small transit agency (fixed route, DRT, integrated trip planning, etc.). Occasionally an agency may wish to obtain multiple types of software in a single procurement, such as integrated trip planning *and* DRT software or DRT software *and* mobile fare payments. This will potentially be more complex than procuring a single type of software. If the agency has limited experience in software procurement, it may be more prudent to conduct separate procurement processes. On the other hand, if the two types of software are known to work effectively together in other settings, the risk is significantly reduced.

Step 3b. Understand Your Available Software Choices

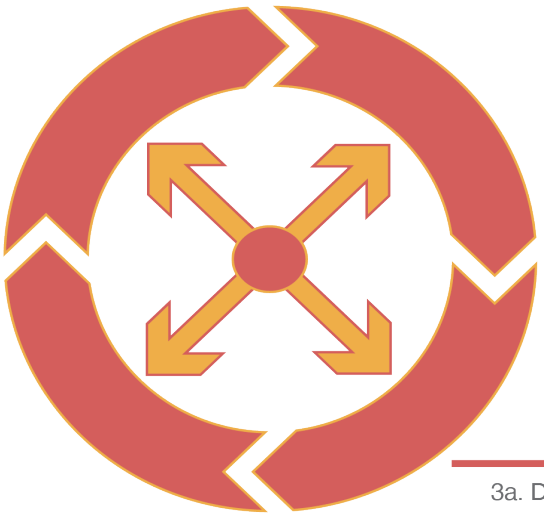
For core software products for DRT or fixed route service management, the number of choices available from commercial software companies is not huge, typically no more than 10 or 12 products (less for fixed route). Agencies can quickly educate themselves about the available products by talking to their peers and other informed organizations—their state DOT, industry associations such as CTAA, the Rural Transit Assistance Program (which is funded by FTA), etc. The latter organizations are unlikely to recommend specific products, but they are generally knowledgeable about the software vendors and their products and can often provide referrals to knowledgeable agencies who are using specific software systems.

Attending a state, regional, or national conference with a focus on small city/rural/tribal transit service is another avenue to obtain information, as many software vendors are likely to be participating in the trade shows that are a normal part of such events. Such conferences are also an ideal forum in which to meet peers who can discuss their experiences with different software systems.

From these multiple resources, the agency can assemble a list of the software companies and their products that seem to be a good fit for its specific needs.

Step 3c. Determine Whether to Obtain a SaaS System or a Licensed Software Product

This is a very important step in the procurement process. The strong trend towards SaaS solutions in the larger software market is much more than a fad, it reflects multiple well-considered business reasons for obtaining software in this manner. Of particular importance, SaaS solutions largely eliminate concerns about the computing platform itself, normally a significant burden for a small transit



Step 3: Move Forward with a Software Product

3a. Determine What Type of Software Your Agency Needs
3b. Understand Your Available Software Choices
3c. Determine Whether to Obtain a SaaS System or a Licensed Software Product
3d. Determine Your Core Requirements for the Software
3e. Develop the Request for Proposals
3f. Evaluate the Proposals and Select the Most Appropriate Software Product
3g. Begin the Software Implementation Process

agency. If an agency perceives that its best available software choices are licensed products, it should then strongly consider procuring a package of software and computing platform hosting, or at least make this an option in the procurement. SaaS and hosted software solutions should only be excluded when broadband data connectivity is not available to the agency.

Step 3d. Determine Your Core Requirements for the Software

As discussed in "Software Requirements, Functions, and Features," the core requirements for the software define what the agency expects it to do. What features and functions do they need to operate their service? Do they need a fully automated scheduling system if they are providing DRT service, or can some mixture of automated and manual scheduling better fit their needs? What administrative functionality is needed in the software? Is an application for drivers part of the system, or does the software only need to include functionality for customers and the agency staff? Does the software system need to provide customers with the ability to directly interact with the software, at least for certain purposes? Does the software need to handle vehicle maintenance and/or the work schedules of drivers? Should the software for the agency staff be web-based, or can it be an application that is resident on a desktop computer or uses a technology such as Citrix to simulate desktop operations?

Relative to the capacity of the software, is the number of concurrent users likely to be a concern? If so, how many concurrent users need to be supported? (If the software is a SaaS application, this is not usually a concern.) Is a SaaS solution desired? If a licensed product is the preference of the agency, should the software be hosted on the customer's premises or should it be hosted in the cloud by a third party (or the software vendor)?

Determining core requirements is an extremely important step in the process, as the decisions made in this step will shape in a decisive way the contents of the Request for Proposals (RFP) document.

Step 3e. Develop the Request for Proposals

This is often the most challenging element of the software procurement process for small transit agencies, as most have little or no experience in actually creating a Request for Proposals (RFP) document. Consequently, most agencies lack knowledge and/or confidence in how to accomplish this. However, there are several alternatives to developing an RFP solely with an agency's own resources.

Peer agencies are often willing to share their RFPs from prior procurements, and since there is typically much commonality in agency software requirements, such "borrowed" RFP documents can often be an excellent basis for the RFP. But agencies should not just copy and paste a document, as the RFP needs to reflect their specific situation, and it is unlikely that their needs will be 100% identical to that of a peer agency.

State DOTs sometimes can provide RFP templates. If the state DOT has conducted a state level qualification process for a certain type of software—and multiple states have done so for DRT software—the RFP or similar document used in that qualification process will often provide a very good starting point for an agency's own RFP.

Model RFPs may also be available from industry trade associations or similar organizations and can provide a strong starting point for an agency.

Consultants can be highly valuable—assuming the agency has an adequate budget to engage one and

task them with developing the RFP. The consultant must be knowledgeable both about the type of service the agency provides and the functional requirements for effective software for that service. But while using a consultant is likely to result in a well-designed RFP, it is very important that the agency engages the consultant at every phase of the procurement process where its own internal knowledge resources are insufficient. It may need to adjust the consultant's scope of work to ensure that their knowledge is available all the way to the end of the process, even if this means a reduced contribution to the development of the actual RFP document. Otherwise, there is a very real danger that the agency will not adequately understand its own procurement document.

For example, there is almost always a question and answer phase to the RFP process, but if the consultant's role does not include this phase it is possible that the agency staff will have great difficulty answering the technical questions—even relatively simple questions—from the software companies interested in submitting proposals. Similarly, if the consultant is not available to assist the agency staff in evaluating the proposals, the agency staff may be out of its depth technically in assessing the relative advantages and disadvantages of different software company proposals.

Irrespective of who is responsible for the actual development of the RFP document, the content of that document represents a significant piece of work and needs to include the following: (1) The RFP must define the desired new software system at a significant level of detail, often including a set of relatively precise specifications for the features and functionality of the software which the agency wants.

(2) The RFP must provide clear instructions and guidance to the proposers about how to describe their software solutions, and what will be their scope of work for implementing the software for the agency.

(3) The RFP must include a structure for the pricing proposal which includes all of the different costs that are associated with making the software operational for the agency and that ensures a fair comparison of different software products.

(4) A proposal evaluation and scoring framework must be created that includes considerations of product functionality, ease of use, the degree to which the product fits with the agency's service delivery approach, and the experience of other agencies with the software product and the software company—as well as the total price of the system.

Step 3f. Evaluate the Proposals and Select the Most Appropriate Software Product

The final step in the RFP process is for the agency's evaluation team to review the proposals and provide a score for each one. This necessitates a careful reading and review of each proposal, with each reviewer typically evaluating the proposal on multiple criteria that were set forth in the RFP document. After summarizing the scores from each of the evaluators, the proposals can be ranked. Sometimes there is strong consensus at this stage that one proposal is clearly superior to the others, and there is no need to interview the proposers or to have them make presentations about their proposal. More typically, the evaluation team will want to have in-depth presentations by the highest scoring candidates. It is typical to have such presentations, which often include product demonstrations, from the 3 top scoring candidates. After the presentations, there is typically another tallying of the scores as the presentation itself will usually be evaluated.

At this point, one company will have the highest score. Some agencies then request the highest ranked proposer to provide them with a Best and Final Offer—a BAFO. The purpose of the BAFO is typically to try to get the highest rated company to reduce its price by at least a modest amount from that which it bid in its proposal. The BAFO phase is totally optional and many agencies do not include this in their evaluation process. Whether a BAFO phase is included or not, at this stage in the process a tentative winner has been determined. Negotiations over the final price could still occur, but the software selection process is essentially completed.

Step 3g. Begin the Software Implementation Process

Once the preferred software provider has been selected and a contract signed, the software implementation process begins. The software company will need to set up the software in either the customer's computing environment or in the hosting environment. It will then engage in a variety of activities to make the software operational and configure it for the agency's specific circumstances. This may also necessitate significant involvement of agency staff.

Following the installation of the software, the next major activity in the implementation process is to train those staff members responsible for using the software in how it works. After the initial training session(s), the staff members will typically continue their learning activities in a less formalized fashion, as achieving true competency in a large software application can require a substantial amount of "hands on" effort.

Key Takeaways

- Determine what type of software your agency needs considering information provided under "Software Functional Types for Small Transit Systems"
- Understand your available software choices by scanning the market for commercial off-the-shelf software (COTS) products that match the software functional type
- Determine whether to obtain a SaaS system or a licensed software product considering information provided under "Software Product Purchasing Options"
- Determine your core requirements for the software considering information provided under "Software Requirements, Functions, and Features"
- Develop the request for proposals (RFP) considering available resources such as model RFPs, past RFPs created by peer agencies, and consultant assistance, keep in mind the RFP will include the core requirements for the software
- Evaluate the proposals and select the most appropriate software product according to multiple criteria set forth in the RFP document
- Begin the software implementation process, considering information provided under "Software 'Ownership' Requirements"

Chapter 4
Step 4: Support the Software

Congratulations! You just acquired a new software system for your transit system. It's not the same as having a new child born, but there are more similarities than you might realize if you want to have a gratifying outcome for what now begins to unfold.

Assuming that your organization followed a good process to acquire the software—such as the one described in Chapter 3—you now have a software system capable of supporting the core requirements of your transit service as well as the needs of the customers who use that service. But if the capabilities of the software are to be of maximum value to your agency, careful consideration needs to be focused on organizing how you move forward with the software. Just as properly raising a child requires dedicated attention and an on-going concern with providing them an environment that maximizes their potential for development and growth, effectively incorporating a software system into the core of an agency's service delivery system requires a similar approach.

General Software Support Needs

Now that you have purchased the software—including training in the software for those employees who will work directly with it as well as a maintenance and support contract—it is *your* responsibility to get full value for your investment. In this chapter, we will provide you with specific guidance about how to do that. But you will be most successful in those endeavors if from the very start you make your software company your partner in this process.

The Importance of a Software Provider Partnership

Supporting the software begins with your relationship with the company that developed the software. This cannot be over-emphasized. Even the best software doesn't always function in the ways that we think it should—minor software glitches or frustrating experiences with widely used software applications that have been around for decades are a fact of everyday life. But with special purpose business software, which is the type of software that you have acquired, the relationship with the company that produces the software is different than when you are a normal consumer. The producer of your software application has a powerful business incentive to be responsive to you.

Your transit agency is one of a small number of organizations—often numbering less than 100, and rarely exceeding 1000—with whom the company that produces your software does business. *Every* one of those customers is typically important, particularly since once they begin using the software company's product they often do so for many years and represent that which is most valuable to every software producer—recurring revenue. No software company wants to lose a single customer if they can avoid it, as they are both difficult to initially acquire in a competitive market and represent a durable revenue stream which can even significantly increase in value over time.

This means that your agency should view its software provider as a partner in your efforts to make your service work as effectively and efficiently as possible. That partnership begins the day you decided to acquire their software and will continue as long as you are sufficiently satisfied with how the software performs to want to continue to use it. It also means that if you begin to experience dissatisfaction with the software, you need to communicate your concerns to the software provider and afford them the opportunity to improve your situation—which is what partners do. The continuation of the partnership is predicated upon them being responsive to your concerns. If they are unable or unwilling to be responsive, it means that it is necessary to find a new software company partner at the earliest feasible point in time.

How do you become a partner with the software provider? It is very simple—first and foremost you communicate with them. And what is the core message that you communicate? That we are in this together. They are the experts in technology and how it can make your operation work more effectively compared to no such technology—and your team are the experts in your services and the specific objectives you have for those services. Your software provider's job is to make the software work as well for you as it can—and your job is to help them understand, at a detailed level, what you want the software to accomplish for your organization. Your

Step 4: Support the Software

4a. Plan for one-time software setup and training
4b. Prepare for ongoing support needs
4c. Consider additional support as the software scope expands

collective task is to learn from each other so that both parties are aligned to *support* your objectives for *your* service(s). This is what software support is really about.

Using the partnership approach as the underlying assumption about your agency's working relationship with the software producer, the steps shown above will, for most agencies, be the focus of their support for the software system. The specifics of each of these steps are discussed in the sections that follow.

Your Software Type has Major Implications for Support Activities

Before we consider the different types of support activities identified above, it is essential that you understand that the *type* of software you have acquired will have a major impact on what the specific focus of your organization's support activities will be. In this context, the "type of software" refers to whether it is intended to be used primarily by your agency's customers—typically referred to as *"customer facing" applications*–or by your agency's staff to manage the operation of your transportation services—referred to for discussion purposes as *"agency facing" applications*.

Customer-facing applications provide important information directly to the trip maker—the latter is the user of the software, not the agency. In most cases, these applications are highly-automated and the user simply interfaces with their computer or mobile device to obtain the information they need and/or to transact with the transit service in some way, such as booking a DRT trip or purchasing a digital bus ticket.

The ride hailing applications of Uber and Lyft are widely used examples of this type of software, as is Google Maps when used for planning a transit trip. For these types of applications, your agency's staff has little or no role in the user process.

Sometimes these applications have very fine-grained functionality. For example, DRT software systems often include very robust manual scheduling and dispatching features that are intended to be used

Customer Facing Software Applications

Customer facing software is designed to be used primarily by trip makers, your customers. The most relevant examples for small transit services are: (1) web-based and/or smartphone-based applications for booking and scheduling DRT trips; and (2) web-based trip planning and schedule adherence applications for fixed route (bus) service.

Agency Facing Software Applications

Agency facing applications are used primarily or exclusively by your transit agency staff members to manage service operations or to assess service outcomes.

only by an experienced scheduler/dispatcher in an operations center. These same systems may also include highly functional trip booking capabilities used by reservations agents. However, the ability of a staff member to handle the reservations element of the software application would provide them with little or no competency vis-à-vis the manual dispatching element of the software. Agency facing applications typically require the users to have specific competency with the different functional components of the software product that they are using.

Of course, some software systems for small transit agencies encompass both agency-focused and customer-focused types of functionality, or the agency uses separate software products that have these different focuses. In such cases, the agency needs to understand the imperatives of software support for both situations.

An example would be a CAD/AVL application for fixed route bus service used by a bus operations manager or dispatcher to monitor the status of vehicle operations. The data collected from wirelessly-enabled GPS devices on the buses is also transmitted to another application designed for the agency's customers—with a customer-facing web-based user interface—that displays information about when a bus will arrive at specific bus stops and if it is on-time. The functionality needed by the customers is simply to have a user-friendly view of accurate information. The agency staff also needs similar functionality, but in addition the software they use will include functionality for taking remedial action if a vehicle is seriously off-schedule or experiencing operational difficulties, and the staff must be competent in using such functionalities appropriately.

Important implications follow from these distinctions between types of software and are discussed in the following sections. In general terms, for software that is primarily customer facing, the most important focus of support is on (1) maintaining high quality data, and (2) ensuring that the software provider is made quickly aware of any issues that customers have with using the application so that rapid improvements can be made. For agency facing software, support priorities will be on achieving the highest possible levels of knowledge about the software's capabilities and a correspondingly high level of competency in use of the software by the agency staff.

Step 4a. Plan for One-Time Software Setup and Training

Purchasing a specific software product is merely the first step in the process of making that software useful to your agency. As soon as the procurement is completed, the partnership with the software provider begins, with 3 activities involving both the agency and the software company typically occurring in relatively rapid succession:

1. Software Deployment

This process involves setting up the software in the computing environment that will be used by the agency, which will be either a remote (off-site) hosting environment or hardware and networking infrastructure located on the agency's premises. While the software provider will be largely responsible for this, the agency may need to be involved, particularly if the software is to be installed on the agency's computing infrastructure.

2. Software Configuration

This process will always include entering data that define the agency's services into the new system, with the types and formats of that data determined by the specific software application. This stage in the software implementation process frequently also includes importing large amounts of data sourced from the prior software system. The most common types of imported data are information on customers and prior customer trips, and data that describe the services themselves (e.g., stop locations for a fixed route transit service).

3. Training the Agency's Staff in How to Use the Software

User training is typically conducted by one or more members of the software provider's staff who are highly experienced in using the software application. Training may involve multiple functional areas in the agency and many different staff members. Multiple training sessions may be necessary depending on the nature of the software and how many functions it includes, as well as the complexity of the software.

It is important to emphasize that user training is a process that begins with training classes organized and led by the software provider, but this is merely the start of the agency achieving competency with the software. In fact, this is the point in the software implementation process where the agency's support for the software truly begins, as the activities until this point have been driven by the software provider and the imperatives of simply making the software operational for the agency. But with the

conclusion of formal training, the agency's internal processes become the primary locus for support of the software, and the issue of agency engagement comes to the forefront.

Agency Engagement

Successful support of a software application, particularly one which is primarily agency facing, requires the active engagement of the agency to fully integrate the software's capabilities into their day-to-day operations. When one visits an agency that has achieved this objective, it is obvious—the capabilities of the software are fused with the capabilities of the people in the organization who utilize it to run the transit services. One observes staff members who use the software as an integral element of their work, and often understand its capabilities at a deep level. They can explain to you not only what it does, but its strengths and weaknesses as well. They often are not simply competent; they are experts in the use of the software.

Not surprisingly, these organizations typically have close, partnership-style relationships with the software provider. Equally unsurprising, they have often used the same software system for many years and have little interest in changing to another software provider's system.

It is commonly said that such agencies have *"taken ownership"* of the software application. For agency focused software in particular, taking ownership is the single most important thing that an organization can do to support their software system. Activities that naturally follow from the taking ownership perspective—ensuring that users are adequately trained and knowledgeable, embedding the use of the software in core agency processes, maintaining good relationships with the software provider so issues can be quickly resolved if they arise—maximize the usefulness of the software to the agency.

Taking ownership by the agency is also important for customer focused software, but the fact that there are many other users of the software than the agency's staff in such situations means that it can at least partially rely on its customers to assess how well the software functions and to alert it to issues. It is always preferrable, of course, for the agency itself to be the entity that is most knowledgeable about its own software, but the reality is that the customers will typically have much more experience with the software than the agency staff in the case of customer-facing software. In these circumstances, taking ownership can mean that the agency regularly reaches out to its customers (via surveys, for example)—or has established on-going focus groups for this purpose—to determine their level of satisfaction with the software and to elicit suggestions for improved functionality.

Step 4b. Prepare for Ongoing Support Needs

As emphasized previously, supporting the software is as much about an agency's orientation towards this technology tool as any specific set of activities. At the same time, there are specific activities that are important to undertake to maximize the value of the software and to minimize the potential for problems to arise.

Impact of SaaS/Hosted Software on Necessary Support Activities

An important distinction for software support activities is between agencies that are using a Software as a Services (SaaS) product or a licensed product that is hosted externally, and those which are using a licensed product that is hosted on the agency's premises (the CPE model). A compelling advantage of the SaaS/hosted product approach is that the agency itself is not responsible for maintenance of the software or keeping it in an operational state at all times. For small transit agencies, supporting a software application is simply much easier and less prone to problems when another organization, which has the resources and the skills to be able to handle this function as a matter of course, is responsible for keeping the software running properly all of the time. Moving the hosting, maintenance, and support function to another organization for whom this is a core business can be a powerful action to improve the support of the system.

Planning and Budgeting for Support Activities

Software support costs money. Not merely the money your agency has already paid—or will pay in upcoming years—for the support and maintenance contract you presumably purchased from your software provider. But also, money to ensure that your staff continues to be adequately trained in how to use the software—since you can anticipate some staff turnover as time goes on and new staff members will need training if their position requires use of the software. Money will also be needed to pay for major upgrades of the software if those are not included in the cost of the original purchase or the maintenance/support contract. If the software system has optional modules that you did not

purchase but could be of value in the future, their potential costs need to be accounted for. The same applies to other types of software that now become relevant as a result of the software that you have just acquired or could enhance the value of that software.

In addition, if the software is "mission-critical"—fundamental to the core transportation service you provide, as is typically the case and almost always applies to DRT software—you may wish to invest in improving your staff's knowledge and competency vis-à-vis the software. Most software producers have annual events—sometimes called user group conferences—at which they provide courses in more advanced learning about their software. Attending the annual event will have a cost but the courses are typically included in the registration fee. Moreover, if your staff attends such an event, they will meet "power users"—individuals highly proficient in the software—from other agencies. Such persons are typically an excellent source of additional knowledge and "tricks of the trade". While sending one or more staff members to attend a user group conference may represent a significant investment for a small transit agency, it often can result in significant dividends, and needs to be seriously considered.

Since the potential costs to the agency associated with support activities such as those cited above have budgetary implications, they must be considered as part of an agency's planning and budgeting process. If an agency purchases software in Year N, and intends to purchase an optional module in 2 years, and also wants to send a staff member to the software provider's user conference in that same year to obtain training and knowledge from other agencies in how to use this new module, it will need to make provision for these additional expenditures in Year N+2. If the agency anticipates it will be too financially constrained to afford these expenditures in Year N+2, it will be necessary for it to develop plans for how to effectively support its transit service in the absence of the optional module in Year N+2.

In general, the agency should develop an annual software support budget that includes all of the relevant potential expenses shown below.

<u>Software Maintenance</u>

If on-going satisfactory operation of the software system is or must be the sole responsibility of the small transit agency—as will be the situation with a licensed software product that operates on the agency's own computing infrastructure—maintenance of the software will be a key on-going activity. As noted in Chapter 3, it may be possible to engage a local technology firm to take on the responsibility for this function, and agencies are encouraged to consider this approach.

The software provider will typically provide its customers with a documented, recommended set of maintenance activities (e.g., shutting down the server once every 7 days and re-starting it to clear the memory of the computer) that should be followed in order to minimize the possibility of problems occurring. Every software application has a different set of recommended/required maintenance activities, often relatively minimal for contemporary hardware and software.

Occasionally software providers will release "patches" to a software application. This occurs when a defect has been discovered in the software that is significant enough to warrant developing a "fix" (a term used more or less synonymously with "patch"). The software provider distributes the patch to its customers (either on a CD it sends to them or via a download from the Internet to the server running the software), who install it and then re-start the software application.

While this process normally goes smoothly, there are

1. Training classes for agency's staff
2. Purchase of major product upgrades (only if agency has licensed software product)
3. Purchase of optional software modules
4. 3rd party technical resources for system maintenance activities (if applicable)
5. Attendance at user group conferences or other conferences relevant to the agency's specific software system.

times when problems occur when installing a patch. For this reason, some organizations are reluctant to install patches, particularly if their system has never manifested the problem that is being fixed. (This will be the case if the patch is related to functionality that the agency never uses.) The philosophy of "if it's not broke, don't fix it" is not necessarily incorrect. However, if by following this approach, i.e., not installing some or all patches, the version of the software that the agency is using begins to deviate significantly from the current version of the product, difficulties can occur when the agency really does need to install a patch. Hence it is important that agencies do follow the software provider's recommendations about maintenance updates to their system.

Software Upgrades

Software providers typically release upgrades to their products periodically, sometimes on a regular cycle. These software upgrades are of two types.

Routine upgrades are similar to the upgrades that occur occasionally with your mobile phone's operating system (or your computer's operating system). Sometimes these upgrades include minor fixes as well. Routine software upgrades make minor improvements in the existing functionality of the software, fix small non-critical defects or imperfections, and occasionally provide minor functionality extensions/enhancements.

Major upgrades typically add significant new functionality to a software product, including in some cases substantial changes in the user interface and the product's work flow. The changes can be of such magnitude that the upgraded software may resemble or behave like a virtually new version of the software product. Using your mobile phone again as a reference point, a major software upgrade can be like moving from the iPhone 6 to the iPhone 10. The two products have much in common, but the iPhone 10 is clearly a different animal than the older model (which Apple continues to sell).

For agencies that use a SaaS product, both routine and major upgrades occur automatically with no action needed by the agency. (This is one of several reasons that small agencies should be seriously considering SaaS products.) The SaaS software provider will inform its customers of such upgrades, and for major upgrades—which are less common than for licensed products, as smaller upgrades typically occur frequently, reducing the need for major upgrades—it will make its customers aware well in advance of the changes that are coming.

For agencies using a licensed product (even when hosted externally), routine upgrades are typically provided at no additional cost beyond that associated with the annual support and maintenance agreement. However, major upgrades typically must be paid for separately, and the cost to the customer may be significant. Every software provider has a different approach to pricing of upgrades.

A major software upgrade can represent an opportunity to obtain a significantly improved version of a software application, particularly when desirable new functionality is included in the upgrade as can often be the case. At the same time, since major upgrades typically need to be purchased separately from the annual maintenance and support contract, the agency will need to make a determination of the merits of the necessary investment. As explained in the Planning and Budgeting section, an agency that has purchased a licensed software product should be allocating funds in its annual budgets to be able to afford the cost of periodic major software upgrades.

If an agency does decide to purchase a major upgrade of the software, it should also allocate resources for additional staff training if the upgrade contains significant new functionality, which will often be the case. This may involve sending one or more staff members to a training class conducted by the software provider. The staff member(s) who have themselves been trained can then train other staff members who use the software.

Additional Software Modules

As discussed previously, many software systems have additional, optional modules that can be purchased either as part of the original acquisition of the core software application or at a future point in time. An important part of supporting your software is to periodically evaluate the functional needs of your service—and your customers functional needs for *using* the service—and determine whether the optional modules from your software provider would be of significant benefit in meeting such needs. If so, your agency should seriously consider purchasing them.

IVR Capabilities for DRT Software:

An Example Optional Software Module

DRT software providers often have optional modules for interactive voice response (IVR) capabilities that enable their core DRT software applications to notify customers via automated telephone calls of upcoming events of interest. These can include: (1) a reminder call the night before a subscription trip that is scheduled to occur the next day; (2) a call that gives the customer a more precise estimate of when they will be picked up than the time window they were provided when they booked the trip; (3) a phone call when the vehicle is only a few minutes away from reaching the customer's location, alerting the customer to the vehicle's imminent arrival.

If customers frequently call an agency's reservation agents or dispatchers to obtain more precise estimates of their pickup time on the day of their trip, an IVR module can be a valuable means of reducing unproductive use of the staff resources of an agency while also providing better service to its customers. It is likely to reduce customer no-shows and late cancellations and increase customer satisfaction. Some SaaS-based DRT software systems provide IVR capabilities as part of their core service offering, but for most licensed DRT products it is an additional, optional module that must be purchased separately.

Step 4c. Consider Additional Support as the Software Scope Expands

The fundamental reason for small transit agencies acquiring and using software is to improve their capabilities for service delivery management and customer inter-action. Typically, those capabilities are limited to discrete services provided by a single transit organization operating in a defined geographic area. For some agencies, however, it can be the case that the scope of their service offerings expand over time, or they begin to engage in activities with other, near-by agencies in pursuit of common objectives.

For example, Mobility Management programs typically involve multiple agencies and making service capabilities available to the public that may be broader than the services provided by a single agency. Or an agency that has been providing fixed route transit and ADA paratransit services determines that it also wishes to provide general public DRT services in all or a portion of its service area.

In such cases, the scope of the original software system will often not be adequate for the needs of the new or planned situation. In some cases, an optional software module linked to the core software application may be able to fill the need, but in other cases the original software system cannot be "stretched" sufficiently to meet the requirements of the new situation.

In these latter situations, the agency is likely to require additional types of software to meet its needs. These software solutions will typically be one of the following:

(1) Relatively broad technology platforms that provide multiple types of functionality, up to and including what is often referred to as Mobility as a Service (MaaS) software. (This category of software includes the trip planning functionality referenced in Chapter 3.)

(2) Mobility Management type of software applications, which are designed to enable different software products of a similar type, but from different software providers, to be able to inter-operate to exchange data and manage transactions that flow

across the organizational and service boundaries of multiple transportation agencies.

(3) Software for an additional mode of service operation for an agency. This could include additional forms of demand responsive services, such as flex-route (sometimes called route deviation) services, which the existing DRT software cannot handle, or a completely different mode such as the addition of DRT service to what had previously been exclusively a fixed route service.

In important ways *each of these situations represent new software procurements* and not more typical software support activities, hence the guidance presented in Chapter 3 about procuring new software is completely relevant and applicable. At the same time, for the first two situations, the primary objective is to extend and expand the capabilities of the agency's *existing software* so that it can support important new agency objectives, and in this sense those situations fall squarely into the software support category.

Moving Towards Inter-Operability, Integration and Platform Capabilities

When a small transit agency needs or desires to significantly expand the capabilities of its existing software to encompass functionalities associated with other software systems, it is moving into a different type of software support situation. Assuming that it is satisfied with the software it is already using, the agency's objectives are likely to include one or both of the following:

(1) Integrating—or at least interfacing—their existing software with a more comprehensive technology platform which can extend the existing software's capabilities, such as with MaaS-like features for integrated trip planning and digital ticketing;

(2) Achieving some form of inter-operability with other software applications so that data can be shared for transactional purposes, e.g., mobile ticketing, handling unscheduled DRT trips from another service provider with insufficient DRT capacity.

Agencies which find that their needs and opportunities are evolving in this direction should be aware that they are on the cutting edge of developments in small city transit services. While relevant and applicable software does exist, there is *very little experience* in actually developing effective Mobility Management programs or MaaS-type services for small city/rural/tribal transit systems. There *are* software companies who have some experience in doing this, but many do not—and the experience base to date is quite limited and primarily involves urbanized area transit services.

National level organizations such as CTAA or RTAP are keenly aware of the opportunities for software technologies to serve as the building blocks for actual implementation of such concepts as Mobility Management and MaaS and have sponsored the creation of resources to disseminate up-to-date knowledge about the state of the art and current possibilities. Some state DOTs (e.g., Minnesota, Michigan among others) are also knowledgeable about this emerging situation and can serve as effective resources.

We recommend that small transit agencies whose next steps of software support activities involve this movement towards more comprehensive software platforms—including some level of data integration and functional inter-operability with other software products—should reach out to external resources to educate themselves about their options. Software for small transit systems is continuing to expand in scope, sophistication and ease of implementation—particularly as a result of the advent of SaaS solutions—and the available capabilities have never been greater.

Adopting a Partnering and Learning Approach

This chapter began with the suggestion that effectively supporting an agency's core software systems is somewhat akin to raising a child from birth. Hopefully the applicability of that metaphor is now more apparent. By creating a partnership with your software provider, achieving a high level of staff competency with the software application that you have purchased, carefully considering how much computing infrastructure support your organization should be providing and how much it should obtain from others, and allocating your financial resources carefully to increase your software's capabilities—and your staff's capabilities to use the software effectively—as your software provider releases major upgrades, your agency can develop a growth and maturation pathway for the integration of your core software with your core service operations. As your agency's technologically-enabled functional capabilities increase and become more

efficient—just as a child's capabilities and scope of competency increase as they grow and develop—more possibilities for effective outcomes for your organization and its customers become manifest.

With the advent of new generation software platforms for integrated trip planning, fully automated DRT services, Mobility as a Service, and mobility management types of initiatives such as the Denver Trip Exchange, small transit agencies can afford themselves of technology capabilities that even 7 or 8 years ago they could not conceive. Moreover, with the rapid adoption of Software as a Service models of software delivery, small agencies can now access such sophisticated technology with far fewer practical complications than has been the case previously. By adopting a partnering and learning approach, a small transit agency can use its necessary software support activities to create a foundation for scope expansion initiatives when those would provide additional functional and customer-serving benefits. This Guidebook is intended to provide you with developmental pathways to more cost-effective and comprehensive service delivery capabilities—which is the ultimate objective in terms of your agency delivering customer benefits. Software clearly has a major role to play in achieving such results and understanding the accomplishment of these objectives as being fundamentally a developmental process—like bringing up baby—can help clarify the needed activities.

Key Takeaways

- Approach the relationship with the software provider as one of partnership
- Plan for one-time software setup and training, which typically includes software deployment, software configuration, and training staff in how to use the software
- Prepare for ongoing support needs such as planning, budgeting, maintenance, and upgrades
- Consider additional support as the software scope expands since many agencies "outgrow" their software over time, adding on additional modules or platforms that may require more integration
- Adopt a partnering and learning approach so you are prepared to expand or modify your software effort as the need and the situation changes

Illustrative Project

The Denver Ride Alliance Trip Exchange is a cutting-edge example of scope expansion. It is a new software-enabled transportation program just implemented in the Denver region that will enable providers of human services transportation (HST) to share capacity and exchange trips. Beginning in 2015, multiple organizations in the Denver region—the Regional Transportation District (RTD, the region's public transit agency), 3 large HST service providers (one of which is part of a city/county government), and a regional scale organization responsible for overseeing human services transportation—embarked on a multi-year effort to develop the technology necessary to achieve the objective of actually accomplishing transactional coordination of human service transportation resources. They secured 2 large federal grants and sponsored the development of the software platform that makes this feasible.

The Trip Exchange software platform makes it possible for agencies—or actually any participating entity—that have trip requests for which they have no available capacity to send the data for such trips from their DRT software systems to the Trip Exchange software platform via automated means. Participating organizations—whether human service agencies, transit agencies or local governments—whose transportation programs have the capacity to fulfill these trips can claim them via the functionalities built into the Trip Exchange software. Such claimed trips are then exported—via automated means—into the claimant agency's DRT software system. In addition, the Trip Exchange software platform includes API's (discussed in Chapter 3) that enable the RTD's FlexRide system (DRT for the general public in 23 service zones) to also participate via its software system. The software platform has been designed so that the RTD's Access-A-Ride (ADA paratransit for the region) system is also technologically capable of being easily connected with the Trip Exchange and this may occur during 2021.

Image source: Regional Transportation District

The HST service providers use a DRT software application from the same large software provider, and that software application has been modified so that it can now interface with the Trip Exchange software. Data for the trips of the HST providers can be transferred back and forth with the Trip Exchange using automated mechanisms. The RTD's own FlexRide services (which use a different DRT software product) can also interface automatically to the Trip Exchange to claim trips under appropriate circumstances.

The Trip Exchange software application itself is an example of public domain software, as the funds for its development (accomplished by a different DRT technology company than the one whose software is used by the HST providers) came from the FTA grants secured by the region. Moreover, its inspiration and original functional model came in important measure from an open source software application developed for an organization in the Portland (OR) region.

Within the Denver region, the precursor to the Trip Exchange system was itself an earlier scope expansion of DRT software applications. The Longmont Coordination System (Longmont is a city of 80,000 population located in the northern portion of the RTD's service district) operated from 2012 to 2019 and enabled the FlexRide program and Via Mobility Services, a large human services and health care transportation service provider, to exchange trips local. A data interchange mechanism was developed by the same two software companies involved in the Trip Exchange project, making it possible for the FlexRide service to utilize Via's excess vehicle capacity during certain hours of the day, and for Via to place ambulatory passengers onto the schedules of the FlexRide vehicles when Via had insufficient capacity.

It is important to emphasize that while the Trip Exchange software itself was newly developed for the Denver region, the ability of the agencies in Denver to interface with the Trip Exchange is a case of pure scope expansion. DRT software already existed for the RTD services and for the HST providers, what was needed was the determination and the resources to extend the capabilities of the DRT—and the DRT software—to be able to cross-utilize available capacity. First the Longmont Coordination System, and then the Trip Exchange, made it possible to achieve this objective.

More details about this project are available on the National Center for Mobility Management's (NCMM) One-Call/One-Click Resource Center, on the Denver metro area spotlight project page.

Worksheets

Instructions

- The intent of the worksheets is to help an agency apply the Guidebook's information to their own situation. It may also help with identifying next steps the agency should take in their software adoption process.
- The worksheets can be completed by the software adoption process lead alone or through collaborative discussions with internal staff and/or partners of the transit agency.
- Review of the Guidebook prior to completing the worksheets is recommended. Certain concepts and phrases are pulled from the Guidebook and may otherwise be unfamiliar to the reader.
- The worksheets can be completed by printing and writing on the worksheets. Feel free to add additional sheets of paper to allow for longer responses.
- If the individual/group completing the worksheets does not have a response for a question on the first pass, then move onto other questions if possible. Afterwards, look into the reasons why the response is not yet clear, and attempt to find the information needed to complete the response.
- If sketching out some of the responses below is preferred by the reader to filling in the blanks, use separate sheets of paper to diagram out the responses. Keep these sketches/ diagrams with the other worksheets as a reference.

Step 1: Set the Software Scope

1a. Clarify the Software's Purpose

Note: Once the response to item 1 is "yes" and the list requested for item 2 is drafted, the agency can move onto item 4.

1. Is it clear which types of software are needed by the agency? Circle yes or no. Review the "Guidebook Focus Areas and Software Types" section of the Introduction and Background Information chapter for examples of software types. For further detail, refer to the "Software Functional Types for Small Transit Systems" section of Chapter 3.

   Yes

   No

2. If "yes" was circled, list each type of software needed by the agency.

3. If “no” was circled, consider two options as potential next steps. First, the agency could hold an internal discussion to pinpoint the types of software, if the software adoption process lead believes this would be a productive option. Second, the N-CATT white paper, a “Framework for Making Successful Technology Decisions,” could be leveraged as resource to help pinpoint the types of software needed through a collaborative and exploratory process. Details on this are available in Chapter 1 and by reviewing the white paper. Write below the next steps the agency intends to take to identify the types of software needed. Once these have been completed, review your responses for items 1 and 2.

1b. Identify General Software Connectivity Needs

Note: Keep this explanation simple, to the best of your ability, and based on currently available information. Review “Inter-Operable Software Considerations: A Short Discourse” within the “Software Functional Types for Small Transit Systems” section of Chapter 3 for more detail.

4. If multiple software types are listed in item 2, explain how each new type should connect with *each other* (e.g., the new “trip planning” app will have a booking option that connects to the new “trip booking” app). If only one software type is listed, leave this blank.

5. For each type of software listed in item 2, explain how it should connect with *existing software* at the agency, if connectivity is needed (e.g., we already have an existing “trip planning” app, it should have a booking option added to connect with the new “trip booking” app).

6. For each type of software listed in item 2, explain how it should connect with potential *future software* at the agency, if any is known (e.g., the new “trip planning” app, to be deployed within the next year, should eventually have a booking option that connects to the “trip booking” app, which we plan to deploy in 3 years.)

1c. Anticipate Resources to Apply to Software Adoption

7. List the financial resources that could be leveraged, now and potentially in the future, for the software adoption process.

8. List the staff resources that could be leveraged, now and potentially in the future, for the software adoption process.

9. List the assets that could be leveraged, now and potentially in the future, for the software adoption process.

10. List the collaborator resources that could be leveraged, now and potentially in the future, for the software adoption process.

Step 2: Collaborate with the Software Stakeholders

2a. Create a Stakeholder Map

Note: A stakeholder map can be a list or a graphical sketch that identifies connections (e.g., cases where the "procurer/manager" stakeholders are the same as the "user" stakeholders). Feel free to sketch on a separate sheet of paper if that is preferred to the list option below.

11. List the "manager and procurer" stakeholders. Include as much detail as you have available such as the name of the individual, their role or title, and the affiliated organization.

12. For the "user" stakeholders, first list the user groups for category identification (e.g., members of the public, agency organizational departments, and others). Then, provide any additional details you have available (e.g., specific members of the public such as app users with visual impairments, specific staff members within the agency's organizational departments, and others).

13. List the "influencer" stakeholders. Include as much detail as you have available such as the name of the individual, their role or title, and the affiliated organization.

2b. Identify Key Topics for Each Stakeholder Group

14. List the key topics for the various "manager and procurer" stakeholders.

15. List the key topics for the “user” stakeholders, taking into account their diversity of interests.

16. List the key topics for the “influencer” stakeholders.

2c. Create a Tailored Information-gathering Process to Integrate Stakeholder Findings

17. Describe the planned meetings and events that would take place for the “manager and procurer” stakeholders. Elaborate on how the findings from the events would be integrated into the software adoption process during Steps 3 and 4 (potentially Step 1 also, if applicable).

18. Describe the planned meetings and events that would take place for the “user” stakeholders. Elaborate on how the findings from the events would be integrated into the software adoption process during Steps 3 and 4 (potentially Step 1 also, if applicable).

19. Describe the planned meetings and events that would take place for the “influencer” stakeholders. Elaborate on how the findings from the events would be integrated into the software adoption process during Steps 3 and 4 (potentially Step 1 also, if applicable).

Step 3: Move Forward with a Software Product

3a. Determine What Type of Software Your Agency Needs

20. Add any additional detail not already included in item 2.

3b. Understand Your Available Software Choices

21. What commercial off-the-shelf software (COTS) products are available that that match the software type, or software types, that your agency needs? List the software companies and their products that seem to be a good fit.

3c. Determine Whether to Obtain a SaaS System or a Licensed Software Product

Note: Refer to “Software Product Purchasing Options” for guidance.

22. Does your agency prefer to have a SaaS system or a licensed software product?

3d. Determine Your Core Requirements for the Software

Note: Only general responses are needed initially. This will help shape your understanding of the software requirements. Consider your responses to items 17-19 to integrate stakeholder findings into the requirements.

23. What features are required, so that the software will meet your agency's needs?

24. What functions are required, so that the software will meet your agency's needs?

3e. Develop the Request for Proposals

25. If you will leverage external resources to help with drafting the request for proposals (RFP) document, what types of resources will you consider? Examples include model RFPs, past RFPs created by peer agencies, and consultant assistance.

3f. Evaluate the Proposals and Select the Most Appropriate Software Product

26. What criteria and considerations would help your agency evaluate the proposals and select the most appropriate software product?

27. Who will be on the proposal evaluation team?

3g. Begin the Software Implementation Process

28. Which software implementation activities do you anticipate being necessary? This could include setting up the software, configuring the software according to the agency's circumstances, training staff members, or other tasks.

Step 4: Support the Software

4a. Plan for One-Time Software Setup and Training

29. Which software deployment activities are likely needed for your situation? Add detail to the response provided for item 28. Which staff members are available to lead or support these activities?

30. Which software configuration activities are likely needed for your situation? Add detail to the response provided for item 28. Which staff members are available to lead or support these activities?

31. Which staff members, as users of the software, should be trained on how to best use the software?

32. In what ways will your agency “take ownership” of the software?

4b. Prepare for Ongoing Support Needs

Note: Consider your responses to items 7-10 to take available resources into account.

33. What items should be included in the agency’s annual software support budget? Are there any particular financial resources to leverage, as provided in item 7?

34. What software maintenance activities are likely to be important? Do these have financial or staff-related implications for the agency? If so, explain.

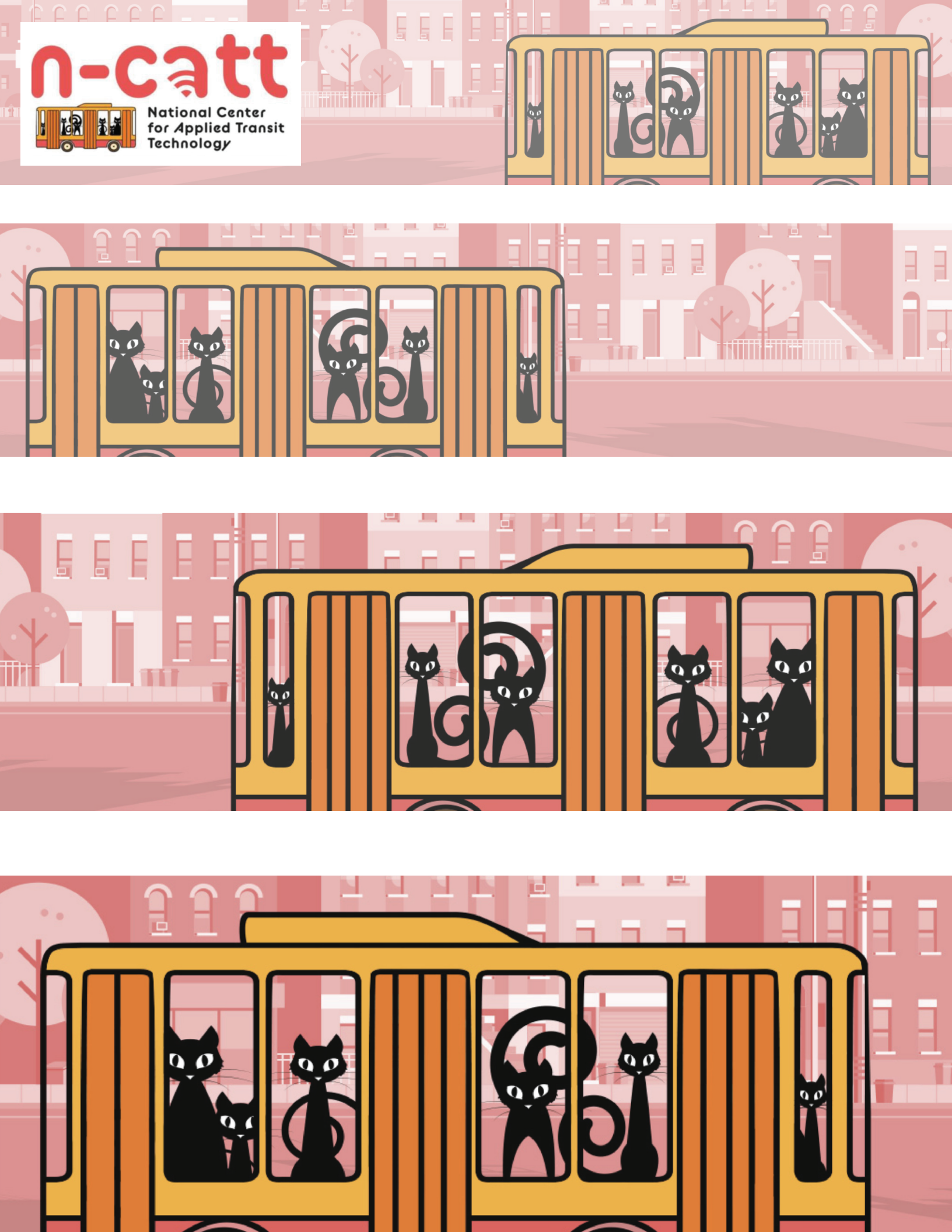
35. What type of routine and major upgrades are likely to be important? Do these have financial or staff-related implications for the agency? If so, explain.

36. Are there additional software modules your agency should consider? If so, list them. Are there any particular financial resources to leverage, as provided in item 7?

4c. Consider Additional Support as the Software Scope Expands

37. Can you, at this point, anticipate any software scope expansion needs over the next 3-5 years? If so, list them.

38. If there are some software scope expansion needs you anticipate, what types of additional support might be needed?


n-catt
National Center
for Applied Transit
Technology