APPENDIX E. PROCUREMENT PRINCIPLES & STRATEGY

E.1 Three Excellent References

The challenges of CVISN system acquisition are not unique to CVISN but rather are common to every area of Intelligent Transportation Systems. Fortunately in the last several years much has been documented, and pioneering procedures and lessons learned are readily available. Three of the references cited in this guide stand out as being profoundly useful in the area of procurement.

The **first is *The Road to Successful ITS Software Acquisition*** [18-20]. If you only have time to read one other document, read its executive summary [18]. The full report [19, 20] is enriched with anecdotes and lessons learned.

The **second is *Innovative Contracting Practices for ITS*** [23, 24]. This is for your team's contracting professional; not warm and fuzzy lessons learned but rather hard cold legal procedures and citations. It is thoroughly researched and footnoted relative to federal/state legislation and case law; comprehensive and detailed; yet readily absorbed. Everything you always wanted to know about ITS contracting. Section II is a primer on federal highway funding. Section III covers types of contracts and methods of award, and barriers and solutions for contracting on ITS programs.

The **third reference is *FHWA Federal-Aid ITS Procurement Regulations and Contracting Options*** [22]. If you have ever had the thought "I want this software developed on a fixed price contract, with a product warranty" then you should read this reference.

Much of this chapter is adapted directly from these three references.

These references are available as a packaged set from FHWA: *ITS Procurement Resource Guide* [9].

E.2 "Software is Different"

Deploying CVISN means deploying multiple software products, and "software is different" [18]. Acquisitions that involve a significant amount of software development are notorious for their problems. Experienced project managers find that proven managerial techniques – which worked so well for them on other types of projects – fail for software. They complain about their lack of insight into what the final system will be like, and their lack of visibility into progress by the contractor.

Representatives from the public and private sectors who have been involved on ITS software acquisitions have very different perceptions as to what goes wrong. Each feels that the other takes advantage of the situation. They perceive that the other party "wins" while they "lose".

This leads to mistrust. Both sides then resort to acquisition practices that further exacerbate the situation.

Compared to traditional state DOT acquisitions, software is different in that it requires a greater customer role than many are used to, from requirements to risk management to system acceptance. As a major consequence, flexibility is needed in the contract to accommodate change and take advantage of the opportunities presented by collaboration and open communication.

E.3 Why do Software Acquisitions So Often Fail?

Reasons given are [19]:

1. Software systems are more complex. They have temporal characteristics, such as sensor data inputs or operator interactions, with real-time interactions. System integration, the most difficult aspect, is primarily a software concern.

2. Human interfaces (including printed reports) are implemented in software. Often the users don't know what they want or don't want until they experience it.

3. On software projects, *production costs* are relatively inexpensive (example: producing a CD-ROM) as compared to *design costs*. This gives the illusion that changes are easy to make. Compare this to highway construction projects, where the fabrication costs serve as a natural impediment to making unrestricted changes to the finished product.

4. Monitoring progress on software projects is difficult. It's not like seeing asphalt being laid down.

5. It is difficult to get a sense of the look and feel for the eventual system from paper documents.

E.4 CVISN Development Contracts Need Flexibility

Some of the decisions you will make are easily reversed. A requirement can be altered with the stroke of a pen. Staffing can be altered with a graceful change in assignments. But by its very nature a contract has permanence through its period of performance. So you want to make sure you are using the right contract vehicle, and with a viable vendor.

No one acquisition practice or contracting mechanism is a panacea that can be relied upon to rescue a project. No acquisition vehicles are ideal for software, **and the familiar highway-oriented engineer/contractor (design-bid-build) approach is particularly inappropriate**. Any type of contract with firm deliverables and fixed ceilings for price does not provide the needed flexibility for building software or modifying existing products.

As compared with other construction projects, software projects need more give-and-take. Flexibility is needed throughout the acquisition lifecycle: in the requirements, in the applied technologies, in the working relationships, and in the contracting mechanism.

Requirements will evolve over time. Reference [18] cites an internal reference that states that two percent of the requirements will change per month. Over just a two-year acquisition this is half of the requirements! Meeting only 80-90% of the perceived needs may be the most realistic and cost-effective solution.

Practical technologies also evolve over time. For example, within the course of recent deployment projects, the World Wide Web evolved from a risky emerging technology to a taken-for-granted feature of the networking infrastructure.

With respect to software development cost, it is impossible to provide precise, reliable cost estimates at the beginning of a project; only a range is possible. When difficulties arise, there must be the flexibility to trade off costs with schedule and functionality as the acquisition unfolds. Here is where the phase-planning approach pays off; it allows the project to be re-scoped. This means you need the flexibility of loosely-defined contract options that are more precisely defined as the project proceeds.

By using incremental phased development, users are more willing to defer capabilities into the next release of the software. On the other hand, if they see a long development cycle, they will try to cram as much capability as possible into the initial release, which further exacerbates the problem. Phased development is discussed fully in *the CVISN Guide to Phase Planning and Tracking* [44].

E.5 Utilize Existing Commercial Software

Purchasing pre-existing ("COTS" for commercial off-the-shelf) products alleviates many of the risks associated with building custom software. Unique requirements can preclude their use, but any such requirements should be examined to determine whether they really are important or whether the system is over-specified. This is an opportunity for your state to partner with neighboring states to develop common CVO business practices and therefore have common requirements.

E.6 Software Acquisition Process

Traditional contracting vehicles used on construction, consulting, and other types of transportation projects are not appropriate for software. Reference [18] makes these key recommendations in the following critical areas:

E.6.1 Acquisition Team

Take a team approach. Skills that must be included on the acquisition team include:

- Software technical experts (system architect; legacy databases; networking).
- Domain experts (IRP, IFTA, safety, inspection, enforcement).
- Contracting officials to select and implement the most appropriate contracting vehicles.
- Legal staff to anticipate and resolve intellectual property rights issues, which are commonplace with software.
- End-users (both within-agency and at carrier's offices).

E.6.2 Requirements Management

The first key activity is to develop a sound set of functional and performance requirements. Unlike other transportation projects, software acquisitions should not develop design specifications or technical requirements at this stage. **Specify "what" not "how".**

An on-going requirements management process will be needed, carried out collaboratively by customer and contractor **with give-and-take on both sides**. There is a compromise line to be walked between freezing the requirements versus never-ending scope creep. It is necessary to allow for extensive iteration between the customer and the designer as part of the system definition. Start off on the right foot with a requirements walk-through with the software contractor, where every requirement is thoroughly examined until the customer and contractor achieve a common understanding of it.

Changes to requirements must be carefully controlled. This does not mean "frozen". There should be sufficient teamwork and contractual flexibility to clarify ambiguities, flesh out lower-level requirements not initially addressed, and reconsider requirements that pose unexpected risk or difficulty.

Human interface requirements are best developed by rapid prototyping and user feedback. It is often impossible to visualize the implication of written requirements. Real-world interaction with the system reveals flaws inherent in the requirements. Users know what they like or don't like when they see it, but often cannot articulate it beforehand.

E.6.3 Intellectual Property Rights

Intellectual property rights come up because of the customer's wish to own and/or retain "the code". Resolving the intellectual property rights must be done before a contract is signed. Chapter 13 of reference [20] provides a checklist to assist with this thorny issue.

E.6.4 Software Scheduling

There are two common flaws with software project scheduling [18]:

- Milestones may be established independent of requirements (for example, to meet political goals).
- Schedules are squeezed so tightly that they are impossible even if everything goes perfectly.

More often "realistic" equates to "pessimistic".

Citing one of its own references, [18] asserts that **one of the most effective ways of lowering the cost and total effort on a software development project is simply to stretch out the schedule**.

E.6.5 The After-Thought of Acceptance Testing

Even though system testing does not take place until much later in the lifecycle, you should plan a formal system acceptance testing strategy early. Pursue acceptance test preparations (preparing test cases, setting up a test environment) in parallel with software development. This will avoid the common problem of treating acceptance testing as an after-thought. In your RFP, ask the vendor to propose their test methodology.

E.6.6 On-Going Management Activities

Conduct monthly and quarterly project reviews; use quantitative measurement data to gain visibility into contractor progress. Have team meetings; the closer the participants are to the day-to-day work, the more often they should meet. Individual development teams typically meet weekly; project teams monthly; oversight committees quarterly. Monitor schedule and inter-project dependencies. Reference [18] emphasized that **if schedule slips occur, do not try to play "catch up"**. Either stretch the remaining schedule in accordance with the slip or reduce functionality in the same proportion.

E.6.7 Risk Management

Engineers define "risk" as probability-of-occurrence times consequences. For risk management to work, there must be an atmosphere that fosters project personnel to come forward. **Continually foster open communications. Each side needs to be able to bring bad news to the other without fear of being "shot down" or of facing recrimination.** This is most effectively done as a teaming activity between customer and contractor since their different perspectives on the system lead them to identify different risks.

E.6.8 Keeping the Customer-Contractor Relationship Win-Win

Open communications allows contractors to better understand customer needs (not "requirements"), while at the same time customers can better understand the implications of their requirements and changes on the level-of-effort required.

Formal *acceptance test plans* clarify what criteria are used to accept a system. This allows contracts to reach closure, and it assures the customer that the system meets requirements and provides sufficient reliability.

Attention to *maintenance and training* clarifies expected roles during these phases. Provisions for contractor maintenance allow bugs to be fixed. Documentation addressing system administration functions alleviate customer dependence on the contractor.

E.7 Software Acquisition Best Practices

The list below of best practices for software acquisition is adapted from [20].

- **Use existing products to the maximum extent practicable.**
- **Build a cross-functional team and collaborate with them.**
- **Maintain on-going, open communications with the contractors and other members of your team.**
- **Document requirements and have them serve as the basis of other activities.**
- **Develop a support strategy for the system.**
- **Trade off requirements to decrease cost and schedule. Keep all three in synch.**
- **Identify problems, record them, and track their status.**
- **Track expenditures and progress.**
- **Conduct risk management, as a team, as an integral part of the acquisition process.**
- **Develop an acceptance test plan and carry it out.**
- **Develop training materials and carry out a training program for use and support of the system.**
- **Have explicit contract language documenting licensing and ownership rights.**
- **Ensure software meets Federal requirements for architecture and standards consistency.**

E.8 Utilize Indefinite Delivery / Indefinite Quantity (ID/IQ) Task Order Contract Vehicles for Information Systems

An ID/IQ contract is one that provides for an indefinite quantity, within stated maximum or minimum limits, of specific supplies or services to be furnished during a fixed period of time, with deliveries to be scheduled by placing task orders with the contractor. See the definitions in Reference [51]. The theory behind ID/IQ contracts is to compete them beforehand based upon qualifications, in response to a broad statement of work. Their flexibility makes them well-suited for phased development and deployment, and amid the rapidly-changing technology underlying transportation information systems.

E.9 Omnibus Contracting Vehicles

A Federal contracting vehicle may become available for State use.

The U.S. DOT Transportation Administrative Service Center manages the Information Technology Omnibus Procurement (ITOP) contracting vehicle. It is a multiple contractor procurement vehicle that is designed to provide a broad range of support resources related to Information Technology. The intention is to extend its availability to state and local government agencies. The second-generation Information Technology Omnibus Procurement (ITOP II) program consists of 26 contractors (and their associated subcontractors) who have formed 35 ITOP II teams:

- Information Systems Engineering (14 contractors).
- Systems Operations and Management (13 contractors).
- Information Systems Security Support Services (8 contractors).

ITOP has these characteristics:

- Indefinite Delivery, Indefinite Quantity (ID/IQ) task order contracts which could total $10 billion over an estimated 7 year life
- Multiple contract types available
  - Firm Fixed Price
  - Cost Plus Fixed Fee
  - Cost Plus Award Fee
  - Time and Materials
  - Fixed Price Award Fee
- Multiple year task orders permissible
- $300 million task order limitation
- Primarily competitive, with some limited directed (sole source) orders allowed
- While primarily services, hardware/software may be acquired and leasing may be included to support full solution based IT contracting

The ITOP Special Project Office (SPO) is responsible for overall contractor performance management. There is a small fee.

E.10Innovative Contracting Practices – Major Findings

In *Innovative Contracting Practices for ITS* [23], several findings cut across all issues as being critical to the success of ITS procurements. These four major findings are:

E.10.1 "How a Procurement is Characterized is Critical."

How one classifies an ITS project is important. For example, procurement rules and regulations may provide much more flexibility to procure financial administration systems than to procure ITS design services. It is important to be flexible in the classification of ITS projects early in the procurement planning process in order to preserve a maximum range of procurement options and implementation strategies.

E.10.2 "Flexible Procurement Practices Work Best if Initiated Early."

Innovative contracting practices can be applied to all phases of an ITS project, but work best if applied at the outset to incorporate strategic objectives into the procurement planning process and the terms of the resulting contracts. Institutional or legal barriers that were identified in advance by participants were eliminated or mitigated by innovative contracting practices.

E.10.3 "ITS Solutions Can Be Implemented at Various Institutional Levels and Project Phases."

State and local transportation agencies implementing federally funded ITS projects have a variety of tools available to overcome contracting barriers to ITS. Not all barriers require legislative or regulatory changes; many can be implemented by flexibly restructuring organizational or managerial aspects of a project. The report identifies a variety of procurement tools to build in flexibility at various institutional levels:

- Partnering with other public and private sector entities.
- Enacting new or revised legislation.
- Selecting funding sources which allow flexibility.
- Leveraging intellectual property rights.
- Utilizing private sector cost sharing with reasonable compliance requirements.
- Carefully segregating, bundling and drafting contract scopes of work.
- Promoting competition among pre-qualified offerors.
- Utilizing evaluation and award criteria that are fair and flexible.
- Incorporating expedited dispute resolution practices.

E.10.4 "ITS Procurements Present Opportunities for Experienced Procurement Professionals to Innovate Within Existing Legal Framework."

Procurement professionals experienced in utilizing innovative contracting practices can assist in removing institutional barriers to ITS deployment. There is however, **a shortage of experienced professionals who are knowledgeable in nontraditional public or private procurement models**. As a result, innovative procurement solutions allowable under current rules, regulations and practices go unidentified, unused, or underutilized. ITS procurements represent opportunities for experienced, creative procurement professionals to develop creative solutions.

Involving experienced procurement professionals early in the planning process enhances a project's chance of success. In addition to in-house professionals, **agencies deploying ITS should consider contracting for external resources to provide innovative procurement expertise**. Having experienced contract professionals involved in a procurement enhances its chances for a successful outcome. The organization from which a contract professional comes is less important than ensuring that a project has access to at least one person who knows the procurement rules, regulations, and practices, and knows how to proactively apply them.

This Page Intentionally Blank