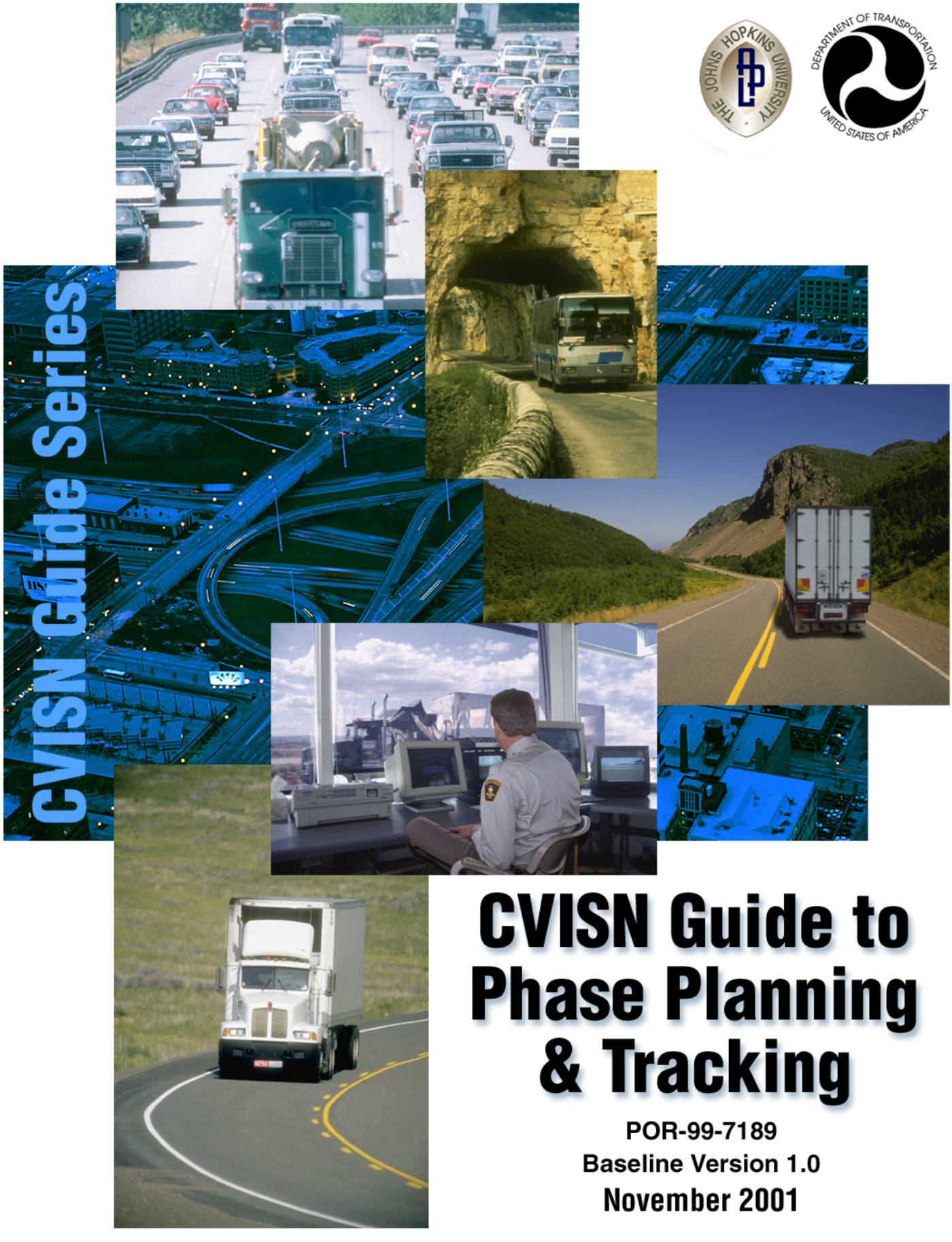


CVISN Guide to Phase Planning & Tracking

POR-99-7189
Baseline Version 1.0
November 2001

CVISN Guide to Phase Planning & Tracking

POR-99-7189

Baseline Version V1.0

November 2001

Authors:
James J. Peter
Valerie B. Barnes

Note

The Motor Carrier Safety Improvement Act was signed into law on December 9, 1999. This act established a new Federal Motor Carrier Safety Administration (FMCSA) within the U.S. Department of Transportation (DOT), effective January 1, 2000. Prior to that, the motor carrier and highway safety program was administered under the Federal Highway Administration (FHWA).

The mission of the FMCSA is to improve truck and commercial passenger carrier safety on our nation's highways through information technology, targeted enforcement, research and technology, outreach, and partnerships. The FMCSA manages the Intelligent Transportation Systems (ITS) / Commercial Vehicle Operations (CVO) program, a voluntary effort involving public and private partnerships that uses information systems, innovative technologies, and business practice re-engineering to improve safety, simplify government administrative systems, and provide savings to states and motor carriers. The FMCSA works closely with the FHWA ITS Joint Program Office (JPO) to ensure the integration and interoperability of ITS/CVO systems with the national ITS program.

Acknowledgements

The authors are grateful for material and inspiration provided by Kim Richeson; for comments and suggestions from John Hardy, Tim Erickson, and Warren Dunham; for reviews by Nancy Magnusson and Mary Stuart; and for production support from PCA Graphics, Theresa Nester, and Beverly Greenwald.

Baseline Issue

This is a baseline document, which has completed internal and external reviews of previously published drafts and preliminary versions. All comments received to date have been incorporated or addressed.

It is disseminated in the interest of information exchange. Johns Hopkins University Applied Physics Laboratory (JHU/APL) assumes no liability for its contents or use thereof. This document does not constitute a standard, specification, or regulation. JHU/APL does not endorse products or manufacturers. Trade and manufacturer's names appear in this report only because they are considered essential to the objective of this document.

This and other CVISN-related documentation are available for review and downloading from the JHU/APL CVISN site on the World Wide Web. The URL for the CVISN site is:
http://www.jhuapl.edu/cvisn/

Please Provide Comments

The authors would appreciate any and all feedback about this document, from modest typos to egregious errors. We are especially interested in hearing from you if you find a topic is missing or if our emphasis is wrong.

Jim Peter
Johns Hopkins Applied Physics Laboratory
11100 Johns Hopkins Rd
Laurel, MD 20723
jim.peter@jhuapl.edu
443-778-5958

CVISN Guide to Phase Planning & Tracking

Table of Contents

1. INTRODUCTION

This Commercial Vehicle Information Systems and Networks (CVISN) Guide to Phase Planning & Tracking shows how to organize and track progress of CVISN projects through their development and deployment.

This is one in a series of guides as shown in Figure 1–1. All guides are available from the CVISN Website [1]. We assume you have already read the *CVISN Guide to Program and Project Planning* [4]. Acronyms are defined in Appendix A of the *Introductory Guide to CVISN* [67] and explained in detail in the *ITS/CVO CVISN Glossary* [71].

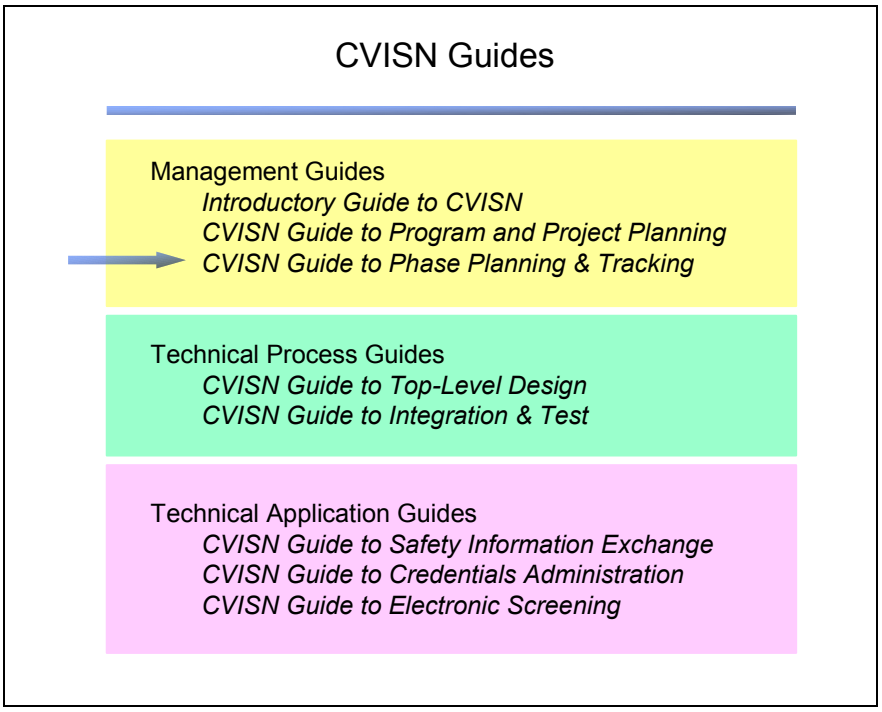


Figure 1-1. CVISN Guides

1.1 Purpose of this Guide

This guide will assist you by distributing the information, knowledge, insight, and experience of others who have traveled a comparable path. It is written for the Program Managers, Project Leaders, Product Technical Leads, and everyone else associated with executing a program or project on a day-to-day basis.

The goals of the Guide are:

- To help you achieve success.
- To make the project's experience a satisfying and career-enhancing one for those involved.

To those ends we define relevant terms, proclaim a few guiding principles, set forth planning and tracking operational concepts, and propose step-by-step tactical processes. We suggest some tools, and finally show examples of the resulting products.

Templates presented in this Guide are available in electronic form from the CVISN Website [1]. Wherever you can, use planning processes, formats, and tools consistent with your organization's existing standards; supplement where there are deficiencies.

1.2 Where Does Phase Planning Fit into the Program Lifecycle?

Recall from Reference [4] the CVISN program lifecycle diagram (Figure 1–2).

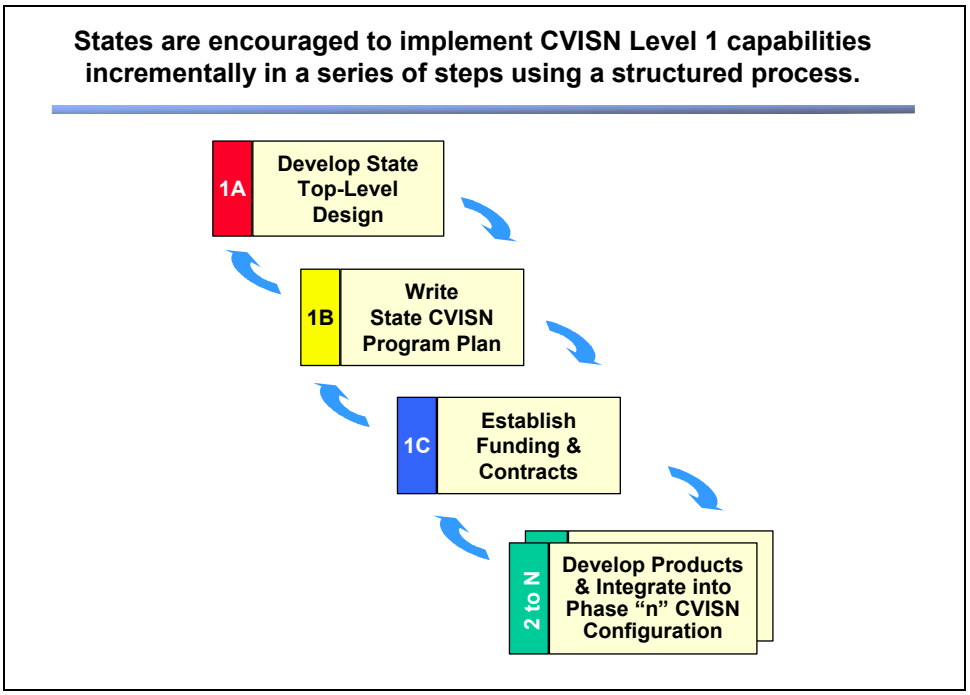


Figure 1-2. Major Steps in Your CVISN Program

You've already completed the steps in 1A, 1B, and 1C. As you begin to approach the first real chunk of development work, you need to put flesh on the bones of the program plan by decomposing the elements of the higher level plan in order to uncover and amplify the details necessary to accomplish the work on a day-to-day basis. This is when you'll do your first round of "phase planning."

Figure 1–3 expands the block shown as "2 to N" in the figure above to explain what occurs in each development / deployment loop.

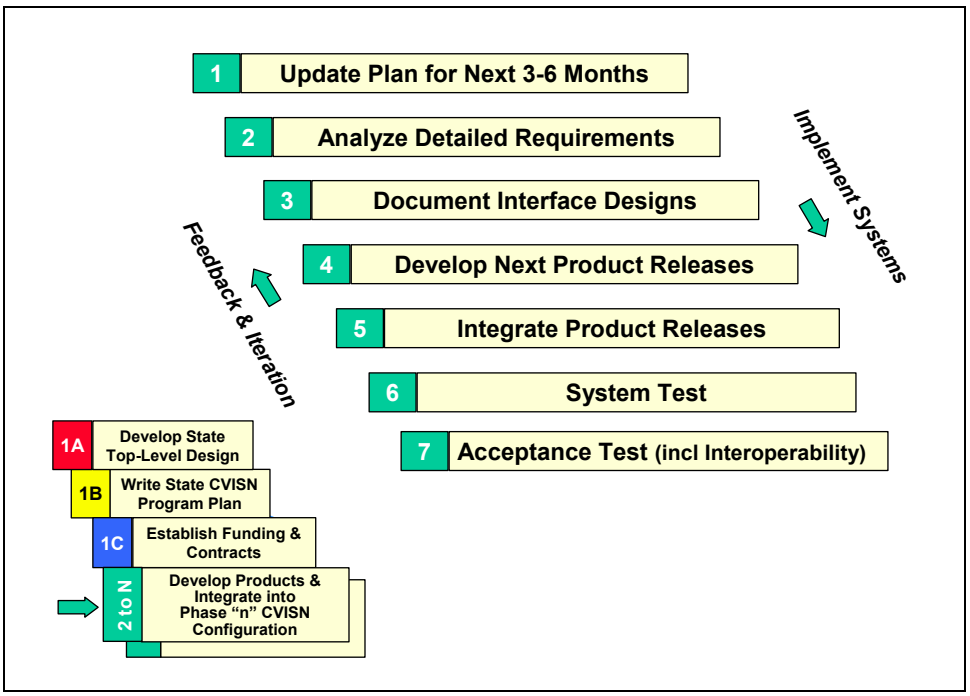


Figure 1-3. The Process for Phases 2 Through N: The Incremental Development Cycle

1.3 What is a Phase?

A phase is a period of time defined for planning purposes to allow incremental delivery of a complex system. Planning and implementing by phases reduces risk: it mitigates the adverse consequences of events such as the loss of a major stakeholder; a subcontractor who never performs and is replaced; or termination of development sooner than expected.

Each planning level (program, project, and product) may have its own phases. Figure 1–4, from the *CVISN Guide to Program and Project Planning* [4], illustrates this concept.

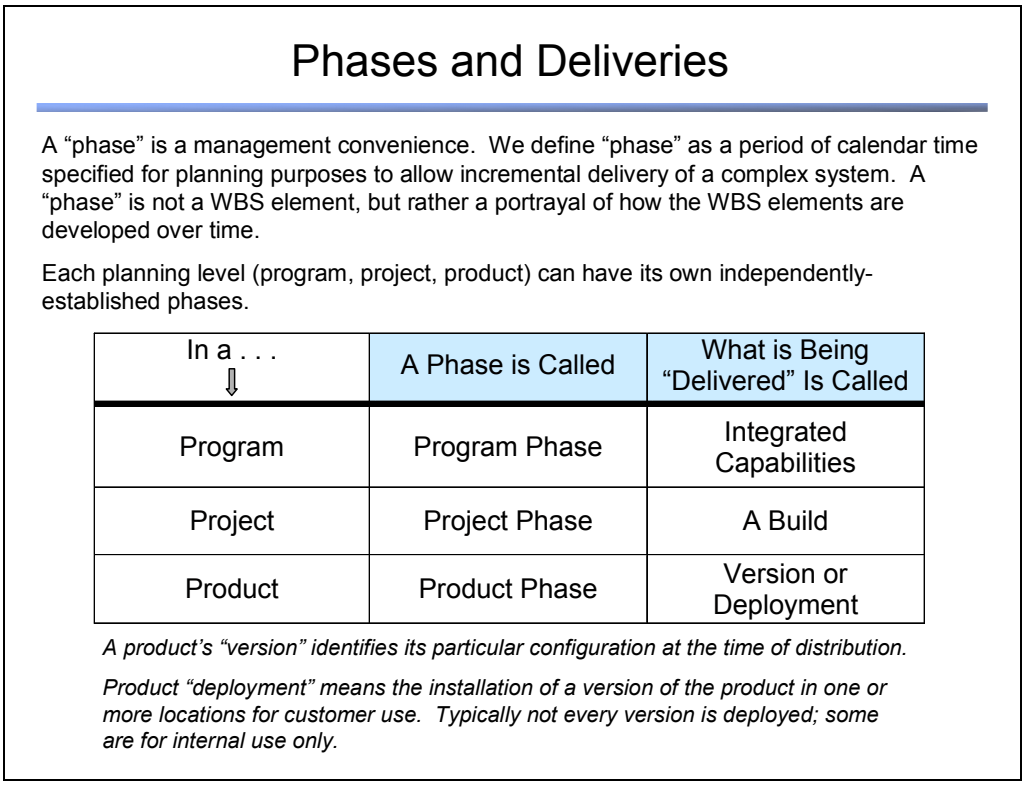


Figure 1-4. Phases and Deliveries

During each phase, products with user-oriented functionality are planned, developed, and released incrementally in a way that subsequent phases can build upon. The phase planning point in the CVISN deployment strategy is flagged in Figure 1–5. The output from the phase planning effort at the beginning of each phase is a Phase Plan; more about what that looks like later.

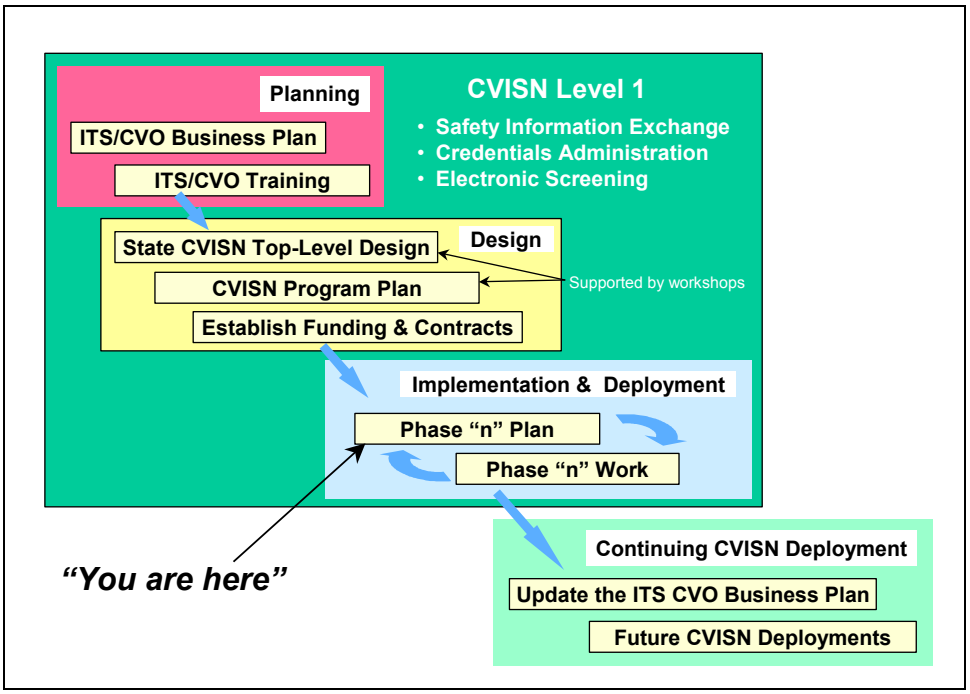


Figure 1-5. Each State Deploys CVISN Capabilities Incrementally Starting with Comprehensive Management and Technical Planning

A CVISN program phase is a roughly three to six month period during which some significant accomplishment occurs in at least one of the three main functional areas of safety information, credentials administration, or electronic screening. The first phase included the program planning (by definition) and is assumed to be complete. Each subsequent phase "n" (that is, 2, 3, 4, 5, ….) focuses on delivering a coordinated set of product versions that incrementally build up to the overall CVISN Level 1 capability. It might be convenient for you to start out assuming that every phase corresponds to a fixed calendar period, say 3, 4, or 6 months. After you've allocated work to phases, you might need to make some phases longer or shorter.

Note that it is not necessary to have significant accomplishments in every project in every program phase, since different projects may be in different stages of development in a given phase. For instance, in your safety project you may start by deploying more ASPEN units to your inspectors in Safety Project Phase 2. In your Credentials Project Phase 2, you may be trying to identify user requirements, or selecting a system developer.

Figure 1–6, portraying project management processes, is adapted from References [12] and [19]. Phase planning and tracking provide an iterative approach to project execution.

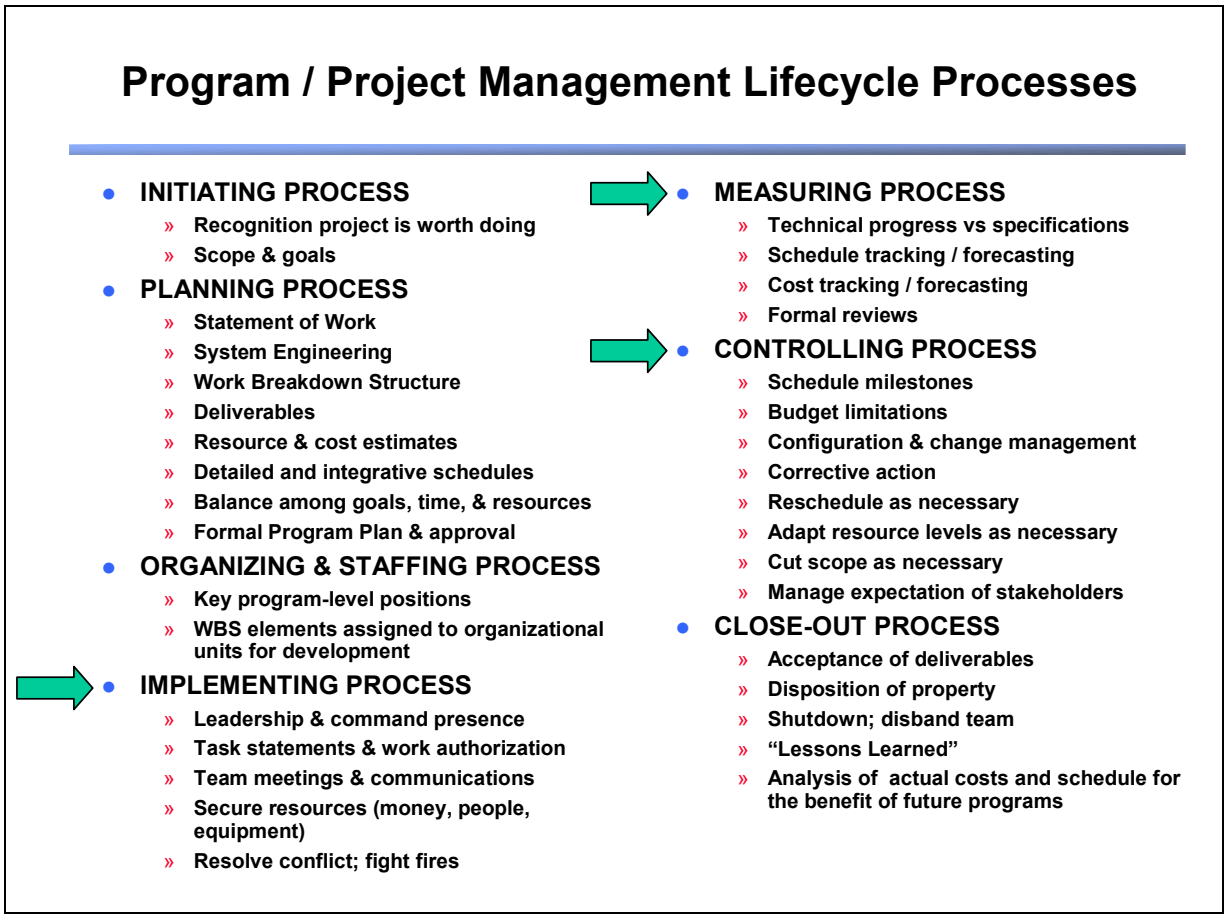


Figure 1-6. Phase Planning & Tracking Provide An Iterative Approach to Project Execution

1.4 What Does a Phase Plan Look Like?

The product of the phase planning effort at the beginning of each phase is a Phase Plan, captured principally in a detailed working-level schedule with resources assigned to each task. Along with that go key charts or tables useful for presenting status.

If the Project Plan is like a mountain climber's base camp, the Phase Plan is like their backpack. Think of the **Phase Plan as those key schedules and charts necessary for everyday work rather than as a bound document.** If you want to be genuinely organized, maintain a Project Leader's Notebook with the latest copies of these key schedules and charts. Figure 1–7 illustrates this concept. A good test of "key-ness" is that every such schedule and chart is just as useful for presenting status as it is for planning. If you make an effort to select and format key schedules and charts so that they are naturally useful for presenting status then, well – they will naturally be used. If they are naturally used, they will be kept up to date. If they are up to date, they will be easy to provide when you are have to present status. As a bonus you will continuously have an up-to-date Phase Plan.

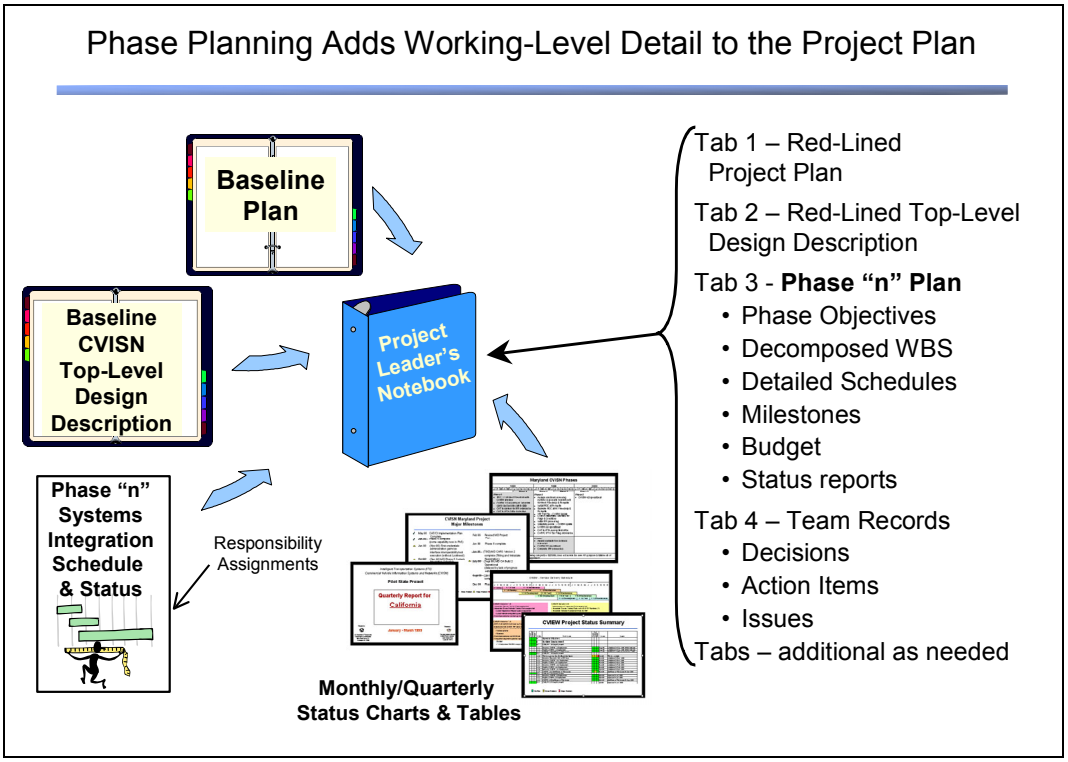


Figure 1-7. Project Leader's Notebook Keeps Up-to-Date Phase Plan Readily at Hand

1.5 We Already Have a Program Plan and Project Plans, Do We Really Need Phase Plans?

A Phase Plan **adds necessary detail** to the higher-level overall project plan, in order to identify, facilitate, and track demonstrable progress. Phase plans are vital because:

- They help maintain focus on objectives.
- When things change, participants can see the impact and adjust to get back on track.
- Without the detail in the phase plan it is difficult to recognize all the work and know that you have actually done everything necessary to successfully complete the project.

Recall that Chapter 1 of the *CVISN Guide to Program and Project Planning* [4] made the distinction between the CVISN **program** as a whole, and its subsidiary deployment **projects**. Phase planning / tracking occurs primarily at the project level, not at the program level, in order to properly focus on very specific activities performed by small teams or individuals.

Each CVISN **Project Leader** develops their own **project Phase Plan**. The technical leaders of the project's relevant products assist. The Phase Plan will show how the project's objectives for

a particular phase will be met, and how the products will be integrated to achieve project-level capabilities. Conversely, the CVISN Program Manager develops the **program phase plan**, drawing upon the subsidiary project Phase Plans to prepare a summary-level view that shows how the projects will be integrated to achieve program-level capabilities; it should be packaged as a presentation to an executive-level audience.

Each Project Leader consults with the Program Manager, the other Project Leaders, and with the line supervisors of the staff assigned to the project. The notion is that phase planning for each project can be done somewhat independently, yet must be coordinated with related projects.

The way project data are summarized into program-level status tends to be very dependent on the preferences and style of the program manager. The phase planning material should be designed to support tracking and status reporting. Keeping the planning and tracking material on-line, for example on a local area network server accessible to all staff, makes it easier to keep the plans and status information up-to-date.

2. INCREMENTAL DEVELOPMENT (SPIRAL PROCESS MODEL)

"All things are a combination of earth, fire, water, and software."
-Adapted from Empedocles, 495-435BC

As explained in the *CVISN Guide to Program and Project Planning* [4], software is different and therefore its development must be managed differently. In the engineering profession this has led to a new incremental product development process model known as the "spiral".

2.1 What is a Development Model?

A product development process model is a mental picture – a conceptual framework that serves as an organizing principle for interrelated activities. The notion of the spiral model as a fundamental principle underlying CVISN phase planning and tracking is so vitally important that we devote a chapter to it.

2.2 Linear Development Model Doesn't Work for Software

The linear model (or "waterfall" model because of its shape when drawn) of product development (see Figure 2–1) worked well for centuries, but fails for today's software-intensive, behaviorally complex systems [10, 11]. Some problems with the linear model are:

- Users don't know precisely what they need from an automated system until they begin to see it in operation, so up-front requirements cannot be adequate.
- There is no opportunity for design re-direction based upon user experience.
- Technology evolves rapidly, thereby making earlier choices less effective or even obsolete.
- The time frame (typically 3 years) is too long from concept to operation, and consequently stakeholder commitment evaporates.
- There is a high risk of never having anything operational.

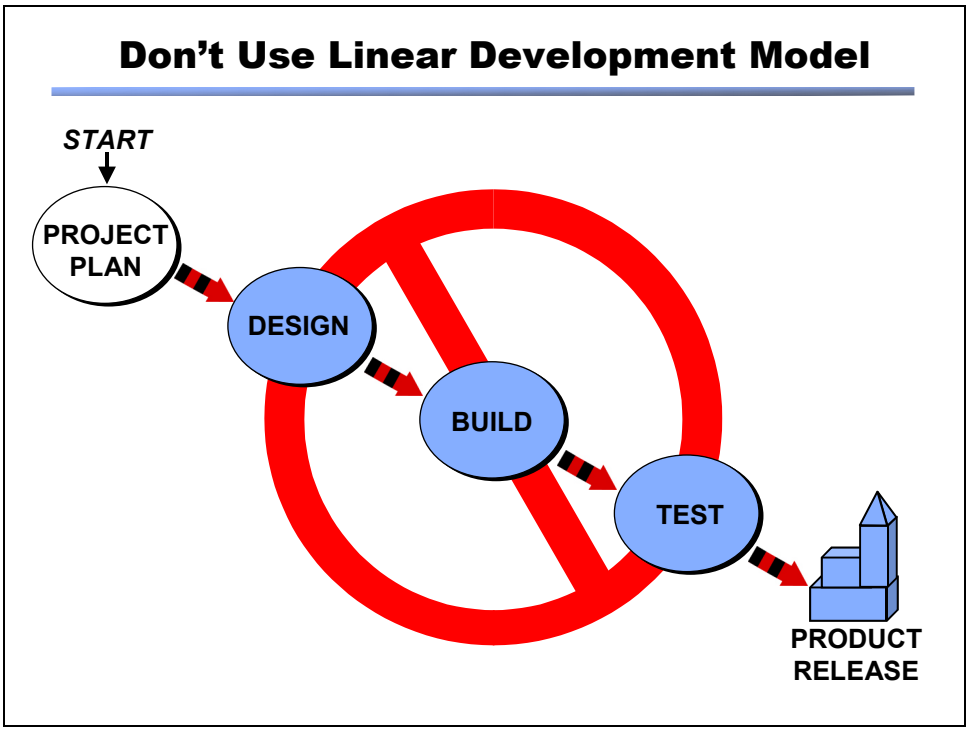


Figure 2-1. Linear Development Model is Inadequate for Software-Intensive Products

2.3 Spiral Development Model Works Well

The essence of the spiral model is first to establish a baseline plan and an overall vision of the architecture; and then to deploy the product incrementally by successive iterations through design, build, test, and next-phase planning.

The spiral development model deals with the shortcomings of the linear development model:

- Users react when they see the system in operation.
- Each turn of the spiral (or phase) is an opportunity for design re-direction based upon user experience.
- Time frame of each turn of the spiral is typically 3-6 months, and therefore stakeholder commitment is nourished by constant progress.
- Each phase is an opportunity to absorb new technology.
- After each turn of the spiral at least that much is operational.

Here's an analogy: suppose your organization began awarding annual bonuses of $15,000 and your family decided to dedicate it each year for ten years to turning your existing house into your dream house. Where do you begin?

The analogy commences in Phase 1 (Figure 2–2) by means of consultations with the architect who prepares a complete site plan. Do you then simultaneously start digging a pool, paneling the family room, and penetrating the exterior wall for a fireplace? NO! What would happen if next year you had to move, or if the bonuses dry up? You'd end up with a big hole in the yard, a half-paneled family room, and a piece of plywood on the exterior wall. Instead you would fully complete one portion of the vision as your first deliverable product for Phase 1 – say the fireplace.

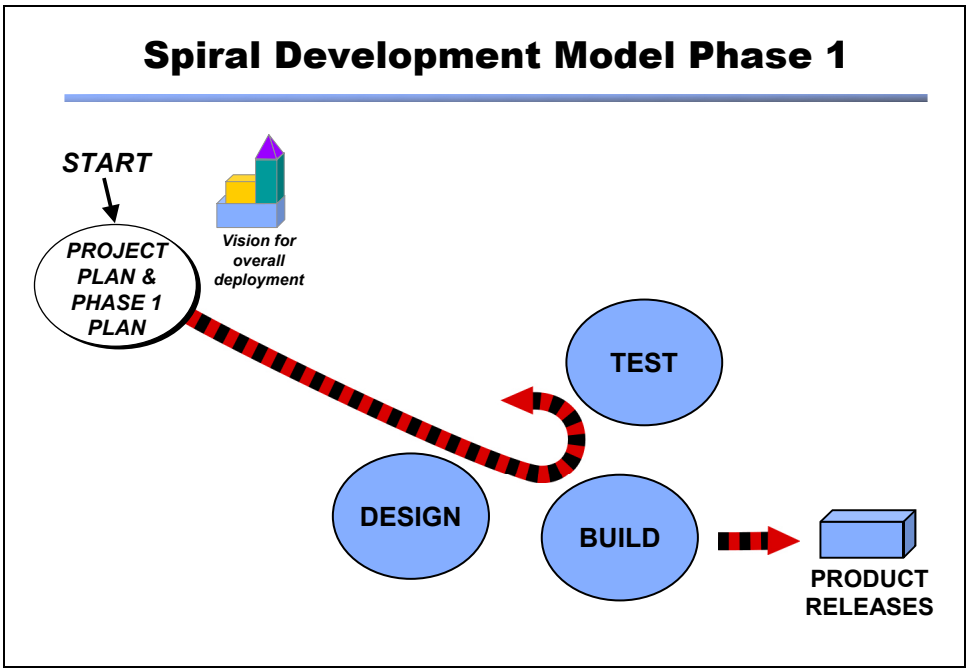


Figure 2-2. First Iteration Through the Spiral Model

For an illustrative CVISN project: at the end of Phase 1 your state architecture has been firmly established, and the essential elements of the computing and networking infrastructure have been procured. You might have hardware (such as servers) but no redundancy yet (such as hot-spare servers); operating systems (such as Windows or Unix); a database management system (such as Oracle or Sybase); network protocols (such as TCP/IP or SNA); and interface standards (such as EDI or XML).

Continuing the dream house analogy: in the second year (Figure 2–3) you are fortunate to receive another bonus. The family's priorities have changed, and now they want the basketball court sooner than the swimming pool. The architect updates the site plan per new safety regulations that now require a fence, and a contractor paves the basketball court. Good thing for you the hole for the pool wasn't already dug, because then there wouldn't have been space for the required fencing.

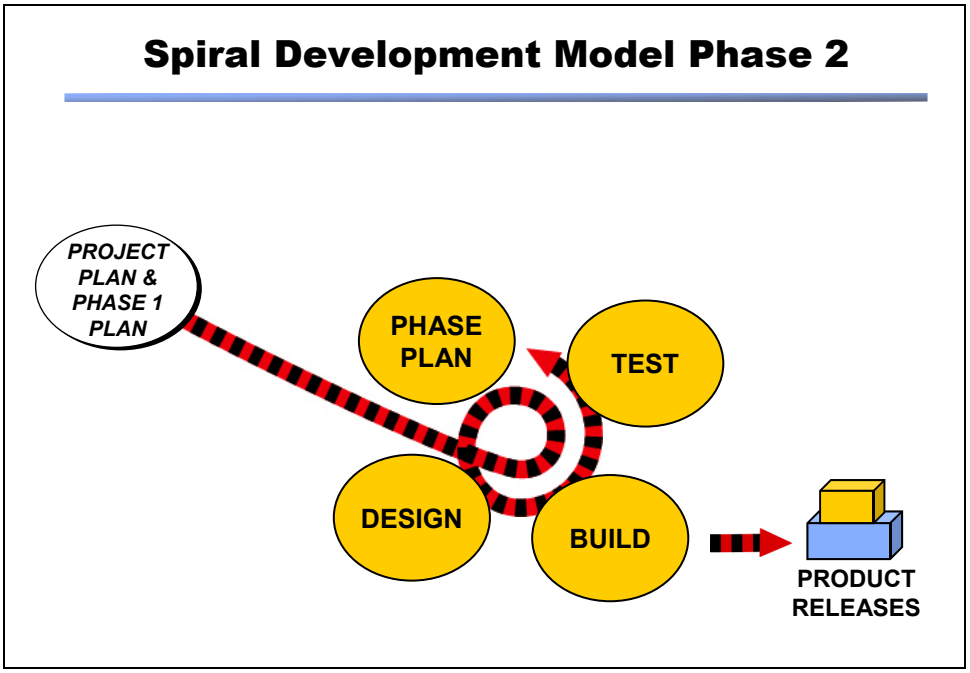


Figure 2-3. Second Iteration Through the Spiral Model

In the illustrative CVISN project: at this point perhaps a prototype user's Carrier Automated Transaction (CAT) software package, along with a prototype state Credentialing Interface (CI) server, would be functional in a laboratory setting – running a few basic user-oriented threads such as the capability to submit an International Registration Plan (IRP) supplemental application but without connection to the IRP processing center.

Continuing with the dream house analogy (Figure 2–4): bad news, bad year – bonuses are much smaller for everyone. You alter the priorities within the constraints, and decide the attic steps go in this year, hopefully the pool next year. Happily the fireplace and basketball court are fully functional.

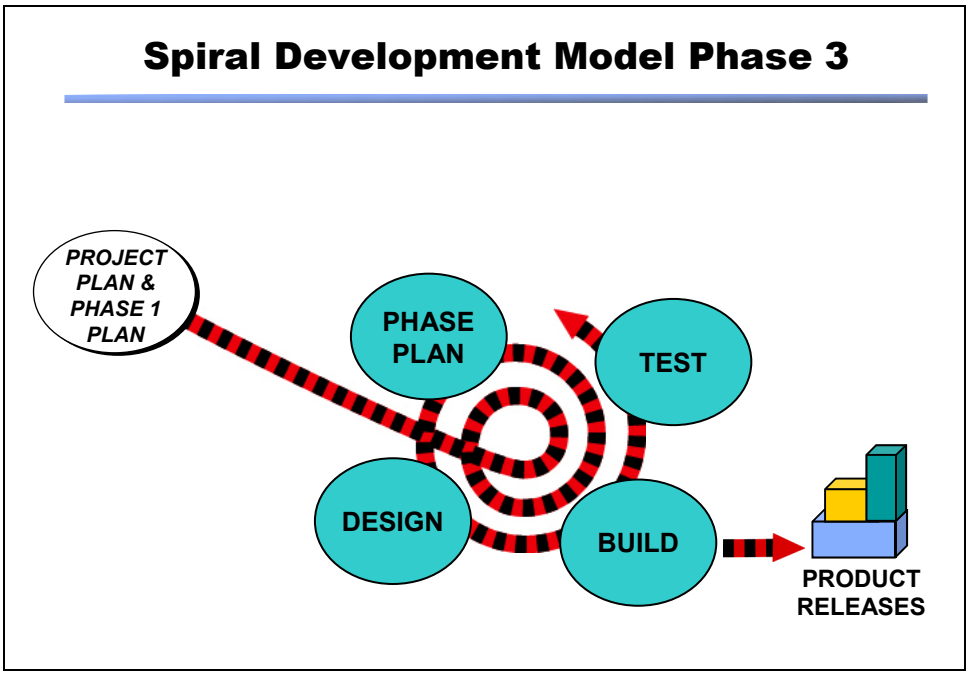


Figure 2-4. Third Iteration Through the Spiral Model

In the illustrative CVISN project: at this point, for example, the CAT hardware would be deployed to a user's site; and the CI hardware deployed in the state's server room with full physical security, firewall, and an uninterruptible power supply. However, there is not yet redundant hardware. Transaction functionality would be demonstrable without embarrassment (such as an end-to-end IRP supplemental application, but perhaps without actually printing the final cab card).

Wrapping up the dream house analogy (Figure 2–5): good news this year, bonuses are higher than ever –but this will be their last year. You complete the pool and fencing, abandon the bowling alley.

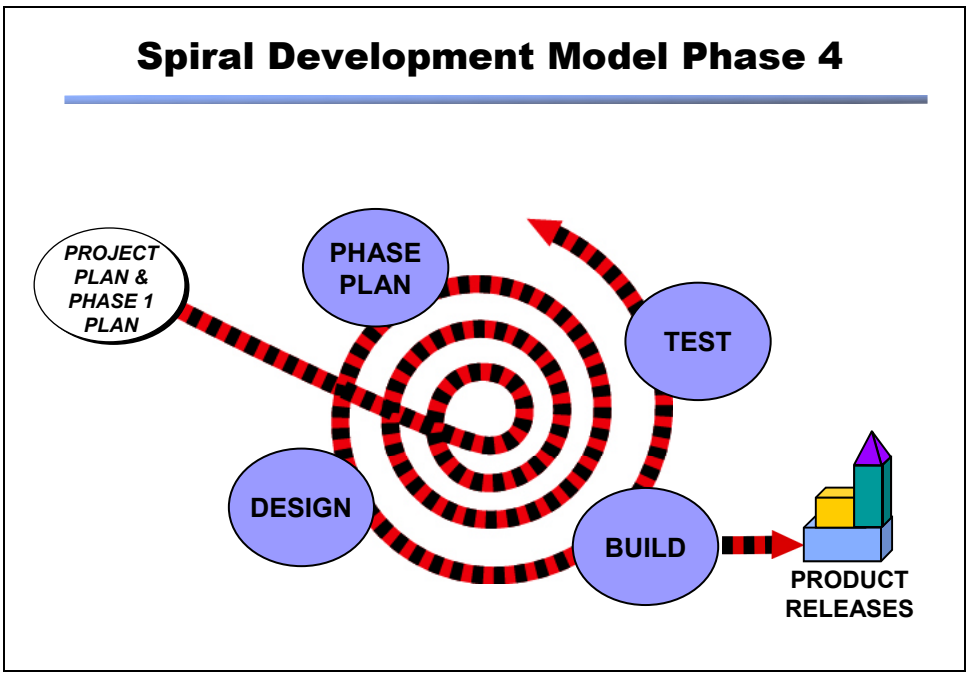


Figure 2-5. Final Iteration Through the Spiral Model

In the illustrative CVISN project: all hardware and infrastructure would be in place, such as a multi-year contract for AAMVAnet and complete redundancy for mission-critical hardware. Complete end-to-end functionality could be demonstrated now, including printing a credential and receiving electronic payment. Perhaps the number of credentials types is still limited, yet the transactions would be end-to-end and useful.

The key point is that after every turn of the spiral there is demonstrable functionality.

In the above approach to development the system is deployed incrementally by successive iterations through design, build (creation), test, and next-phase planning. Detailed requirements analysis and design occur in each phase. This carries the risk that some "hard" requirement is identified at the detailed level that cannot be accommodated by the baseline top-level design. The program-level Configuration Control Board should address such problems. The advantage of the spiral approach is that some capability is available at the end of each phase, and end-users see what the developers are going to provide. That early end-user evaluation allows their feedback to influence future phases. Also, in each phase the designers have an opportunity to take advantage of new technologies.

2.4 What is a Software "Build"?

In the software development world "build" is not only a verb but also a noun used to describe a software entity. The *IEEE Standard Glossary of Software Engineering Terminology* [23] defines a "build" as **an operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide**.

The notion of builds is useful for phased development and deployment, but usage requires rigorous record-keeping in order to know exactly what is "out there". In the sections below we consider a "build" as an integration of particular versions of products into a working system.

2.5 Configuration Management During Phases and Builds

Figure 2–6 shows the relationships among program phases, project builds, and product versions. Each program phase is associated with a set of defined operational builds within each project. Each project's build is a set of specific versions of products ("releases") that have been integrated and tested together, to provide specified capabilities for that project. Every product release has a version number and an associated description. All of the preceding are under configuration management.

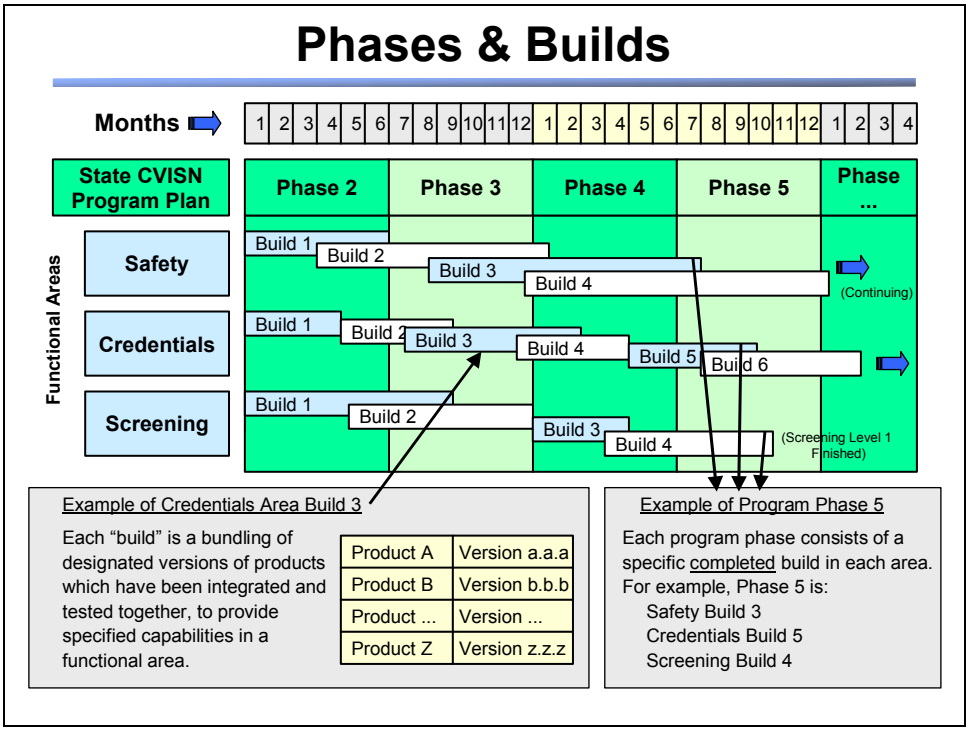


Figure 2-6. Relationships Among Phases, Builds, and Product Versions

A rigorous configuration management process notifies all concerned parties how to handle change requests and approvals, and how to control hardware modifications and software versions. Configuration management is a discipline applying technical and administrative direction and surveillance to identify and document characteristics of every item designated to be under configuration control.

As you make incremental deliveries of products, you need to be able to identify all the components in each delivery, and keep track of problems detected in one phase so that you can control how and when proposed resolutions are implemented.

Figure 2–7 shows how individual products might be identified. Releasing a product version usually means making it available to users for testing or operations. Keeping track of versions of different products makes it possible to identify what components are used to achieve each build's capabilities within a project (Safety Project is shown in this example.) Be sure to establish some naming convention for versions that everyone understands and can apply consistently.

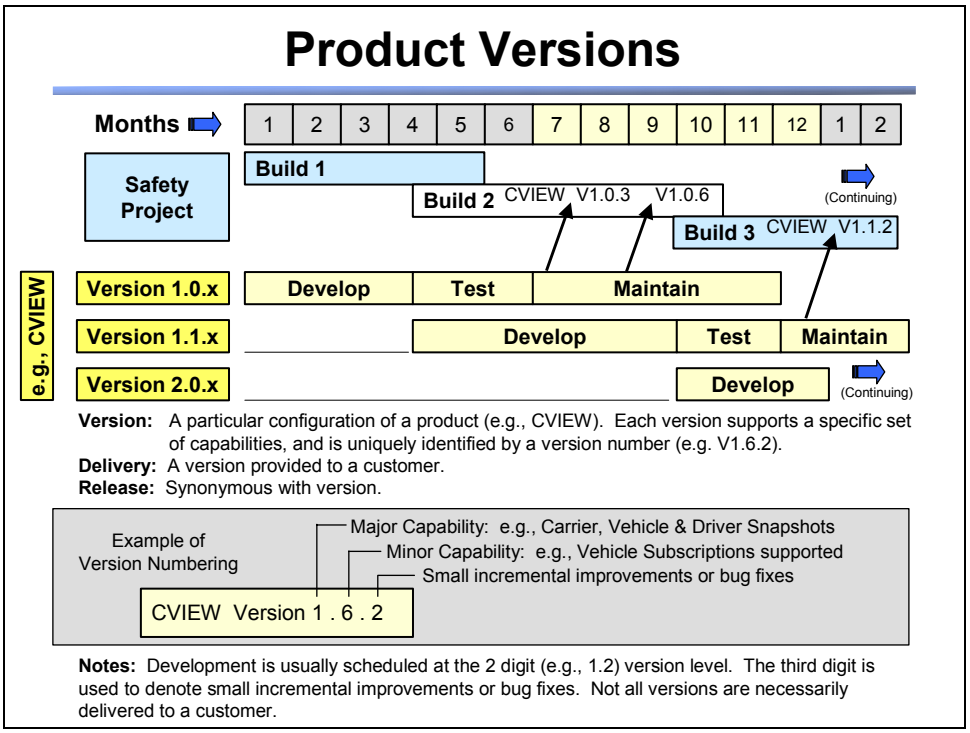


Figure 2-7. Version Identification is Part of Configuration Management

The *IEEE Standard Glossary of Software Engineering Terminology* [23] defines **"version"** as **a release of a computer software configuration item –** meaning that the version number uniquely identifies a configuration-controlled software entity and distinguishes it from similar versions released earlier or later than the one you hold in your hand. This matters when you need to know exactly what the software is supposed to do or when you are calling the manufacturer for help. Figure 2–7 is consistent with that published standard; however you are likely to encounter different usages by various vendors and within other organizations.

By their nature **these terms are used recursively**: one person's system (top-level deliverable) is another person's component (minor element of a larger deliverable system). There can be builds of versions of builds of versions. Although it may seem potentially confusing, it is easily resolved if you explicitly point out the level in the product hierarchy you are referring to.

This Page Intentionally Blank

3. SCHEDULING – THE MOST IMPORTANT PHASE PLANNING ACTIVITY

3.1 "Project Management Software" – Does it Really Do That?

Not really – it doesn't do any "management" – that takes a talented human like you! A better-named category would be "Project Scheduling and Tracking Software". What it does do is figure out important error signals (as a feedback control system does) that the Project Leader can then use to strategize project interventions.

In this Guide we will say "scheduling software package". This important class of desktop software is a combination of database, spreadsheet, and graphics capabilities: database because of the structured information retained for each task (or "activity"); spreadsheet because of the cost calculations it can perform; and graphics because of the extensive large-format printing capabilities.

In the same way that a word processing software package takes a lot of the pain out of writing by easing the manipulation of text, the scheduling software package takes a lot of the pain out of scheduling by easing the manipulation and linking of tasks. That makes it cost-effective for staff to regularly update schedule information. Up-to-date schedule information is a valuable asset.

Developing an achievable, detailed schedule for work occurring over the next 3–6 months is arguably the most important activity performed during phase planning. That's why we put this chapter right here.

3.2 Basic Capabilities of a Scheduling Software Package

A scheduling software package facilitates your preparing a plan for allocating the resource of time. It associates the use of time with achievement of goals. The project's Work Breakdown Structure (WBS) provides the framework for schedule traceability – that is, from master schedule to detailed schedule; and from one deliverable to another. (See Chapter 4 of the *CVISN Guide to Program and Project Planning* [4] if necessary to refresh yourself on WBS concepts.)

Scheduling software requires that you enter at a minimum:

- Task descriptions (about 60 characters maximum).
- Task durations (normally in days, between about 1 and 30).
- Task inter-dependency relationships (normally assumed finish-to-start).

It is stressful for people to estimate task durations. Therefore, you will have to set a scenario for them. Tell them to assume that they can work on any task for half a day each day, uninterrupted. Then double the duration. This will compensate for natural over-optimism, and the interference of other work assignments. Team leaders will also find it useful to identify named resources (example: people, equipment) for each task, in order to prevent cumulative overloading.

3.3 What's the Payoff for all this Effort?

Scheduling is a chore – especially for the first several weeks while everyone struggles to articulate, understand, and clarify – but the payoff is definitely worth it. The most significant benefit is that the team actually sat down and thought through everything they have to do. Be patient, it will take weeks or even months for the schedule to gel.

Once the input data is stable, the software package will produce handsome large-format printouts such as a critical path network diagram and a Gantt chart. (Samples are forthcoming in Section 3.5.) The network view emphasizes task inter-dependencies. The Gantt chart view emphasizes task durations and calendar dates, and it can be filtered according to selection criteria – for example, to show only the activities to be performed by any one person or organization. Once named resources have been entered the software package can also produce resource utilization charts, and even help via automatic resource leveling.

To fully take advantage of the capabilities of this software package you will need access to a wide-format color ink-jet printer (or “plotter”) that can print on 36-inch wide paper. Such devices are normally deployed as shared resources on your local area network. If you don’t already have a large-format printer add it to your capital planning list; fortunately the price of such devices has come down by an order magnitude to just a few thousand dollars.

3.4 Critical Path Network Scheduling Computations

Arguably the greatest benefit from this software comes from capturing the dependencies among activities; that is, which activities have to finish before other ones can begin. The conventional arrow notation shown in Figure 3–1 means that Task A has to complete before Task B can start.

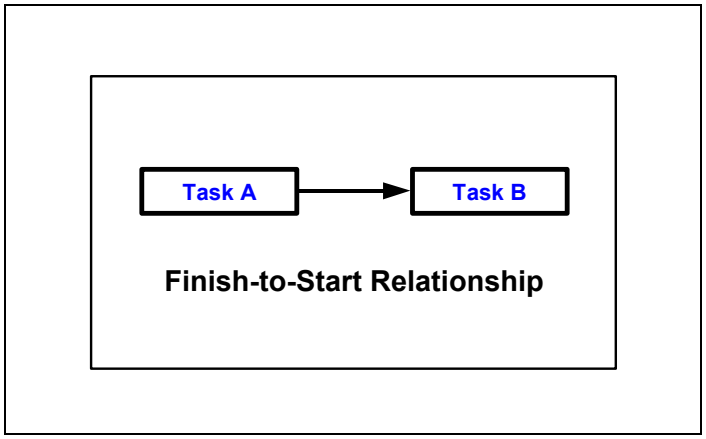


Figure 3-1. Activity Dependency Notation

Task inter-dependencies are best examined in the *activity network* view, where relationships among tasks are shown without regard to a calendar axis. Once you capture task inter-dependencies then the software package can compute the "critical path", which is the string of activities with zero "slack". Slack is defined as the work time that an activity may slip without delaying the end of the project. By definition, all activities not on the critical path have slack; it comes about because they are in parallel with activities that take longer. Adding more activities to a schedule doesn't necessarily mean a later project end date so long as they are not on the critical path, and are not performed by the same resources.

An activity network models the realities and constraints of the development and deployment process and is therefore the most effective framework for planning, scheduling, monitoring, and controlling detailed activities. A simple example of an activity network is given in Figure 3–2. People use these terms synonymously to describe it: bubble chart, CPM (Critical Path Method) diagram, Critical Path Network diagram, and PERT (Program Evaluation Review Technique) chart.

A *Gantt chart* generated for the same tasks is given in Figure 3–3. The Gantt chart shows the schedule as bars spanning calendar time. Scheduling software packages make it possible for you to easily switch from one view of the schedule to another. It is also possible to capture task inter-dependencies in the Gantt chart view (start to finish, start to start, etc.) and depict those linkages as vertical lines between activity bars.

Scheduling View: Activity Network

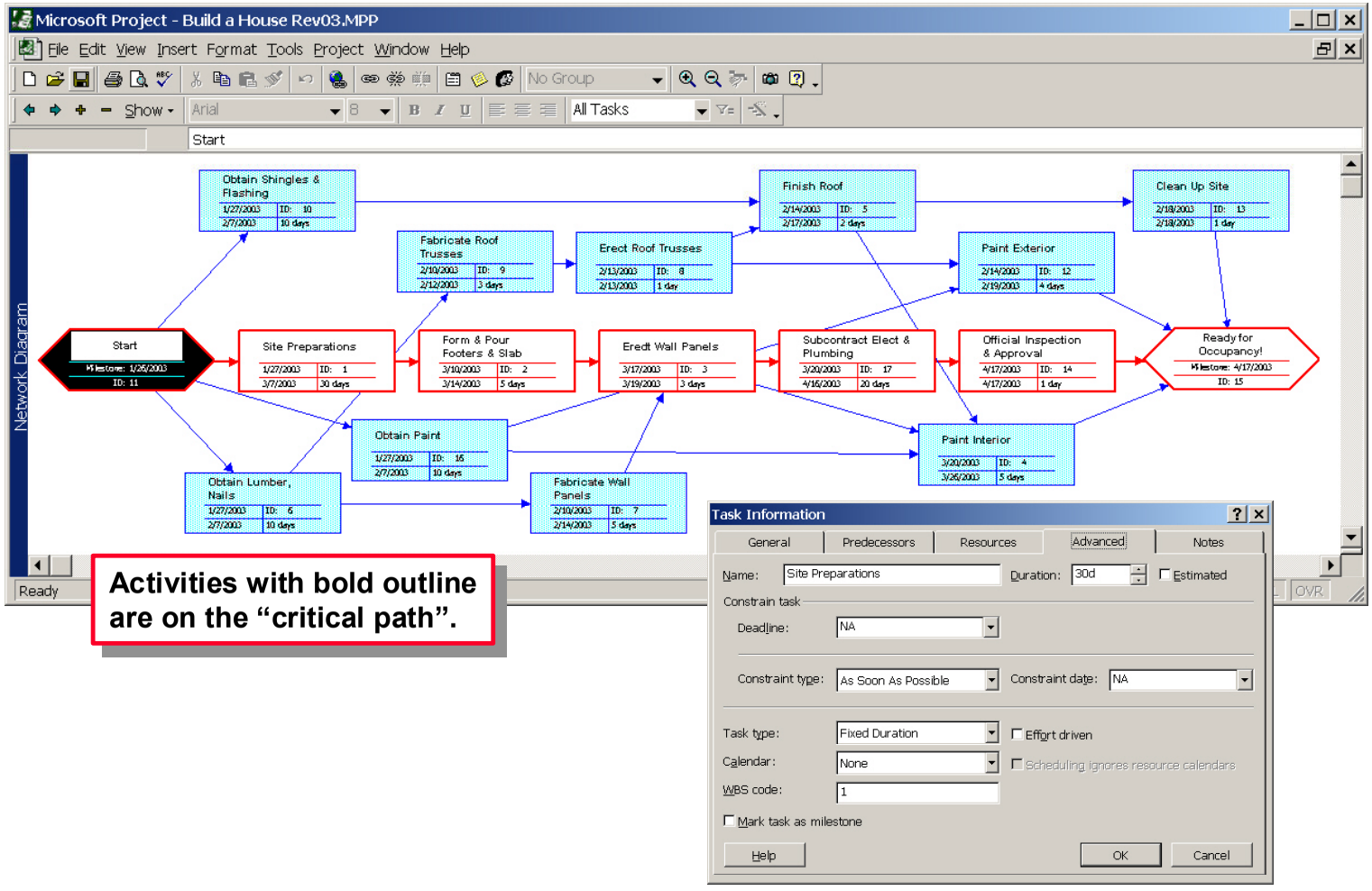


Figure 3-2. Activity Network View is Best for Showing Task Interdependencies

Scheduling View: Gantt Chart

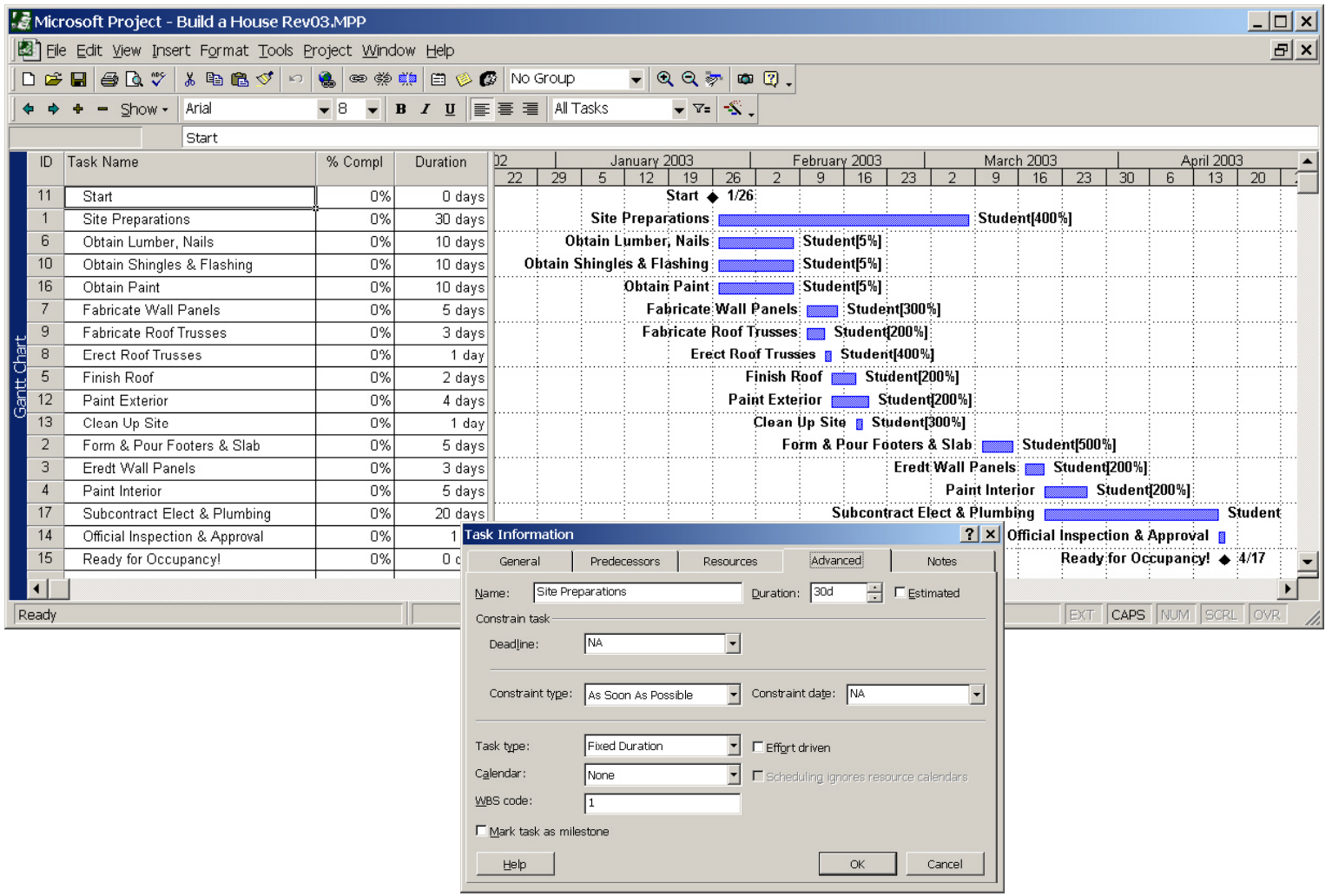


Figure 3-3. Gantt Chart View is Best for Showing Tasks Versus Calendar Time

3.5 Relationship of Product Schedule to Integration Schedule

Resist the urge to toss the detailed schedule for every product into one gigantic schedule for the whole project or program. Instead, maintain the individual schedule computer files for each product, with say 100-200 tasks in each. Create a stand-alone higher-level activity network that shows the interrelationships among products and how they come together for integration, test, and deployment as illustrated in Figure 3–4. Include the preparation of test data, documentation, and training materials in addition to the products themselves.

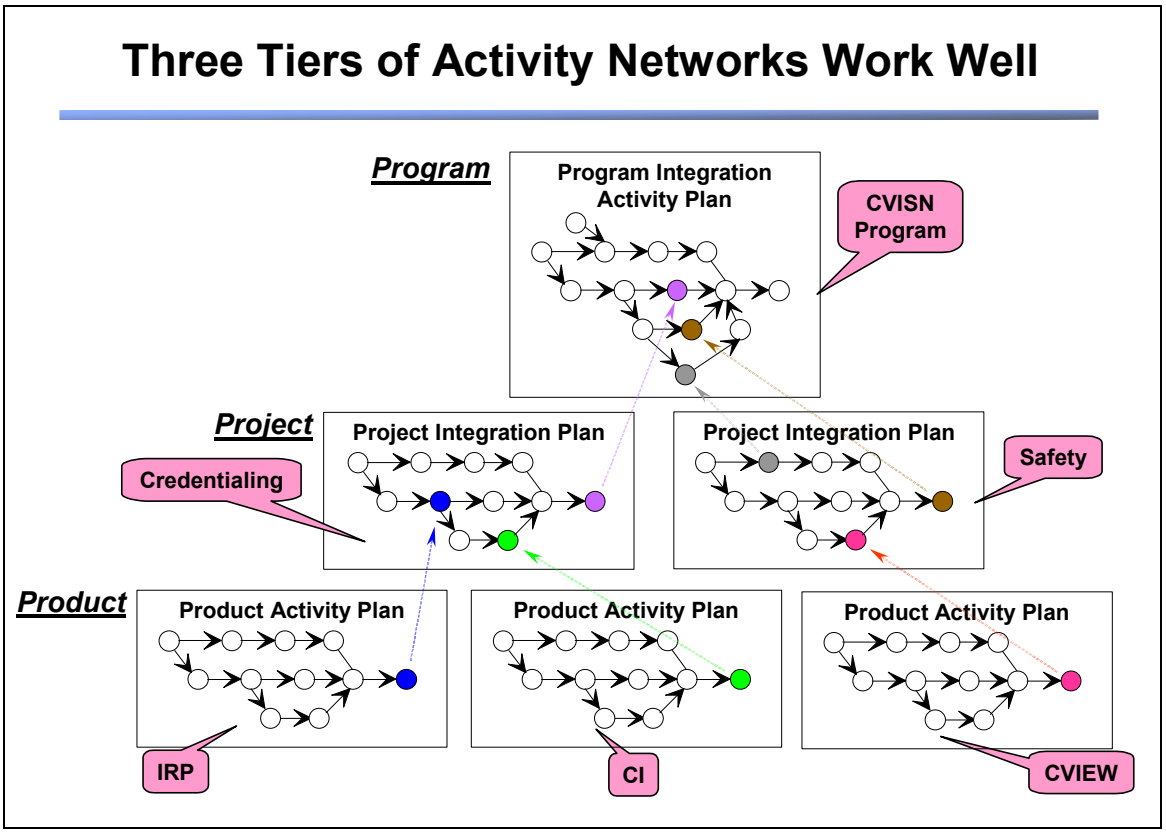


Figure 3-4. Don't Toss Everything Into One Giant Schedule

At the very highest level you can create a stand-alone activity network that shows the key interrelationships among project activity networks, to portray how the overall program will be integrated and delivered.

3.6 Advanced Scheduling Software Capabilities

You are probably familiar with spreadsheet programs and how the values for cells can be linked among separate files. Similarly, modern scheduling software packages can portray dependency links between tasks in different files. They can also handle "subprojects" – detached relatively small projects each of which is subsumed into one task in a higher level project.

This Page Intentionally Blank

4. CVISN PHASE PLANNING

Amid a wash of paper,
a small number of documents become critical pivots
around which every project's management revolves.

The above from Rechtin [10] quoting Frederick P. Brooks's classic *The Mythical Man-Month*.

A systematic planning process is essential to reveal and formalize the work to be done, and then to communicate the results among team members and other stakeholders.

At this point in the project management lifecycle you have moved from the up-front planning stage to the implementing stage. Detailed task planning and tracking are mutually coupled and interdependent. In this chapter we'll focus on phase planning, a "lean and mean" effort to produce a concise **Phase Plan**.

As discussed in Chapter 2, CVISN implementation follows the spiral development model via successive phases of incremental delivery. In this chapter we first discuss some **guiding principles** on which our recommended CVISN phase planning process is based. Next we discuss **operational concepts** that are generalized from particular instances or experience and can be applied to new situations. Lastly, we lay out phase planning **process steps** which you should tailor to your unique project environment, organizational culture, and available tools.

4.1 Phase Planning Guiding Principles

Figure 4–1 lists the principles for phase planning in CVISN.

1. **The spiral development model applies to the system development in CVISN**. This is a re-statement of Chapter 2. As explained there, the essence of the spiral model is to first establish a baseline plan and an overall vision of the design, and then to deploy the products incrementally by successive iterations through design, build, test, and next-phase planning.

2. **Incremental deliveries result in incrementally improved operational practices.** Planning, developing, and releasing products incrementally – such that useful end-to-end functionality is delivered with each system build or product release – enables end-users to do their jobs better with each iteration. This principle suggests that we should aim for increasing end-to-end functionality as opposed to increasing stand-alone functionality whenever possible. For example, one IRP transaction type that goes all the way through the back-end record keeping system is preferable to a user screen that shows all the transaction types but doesn't complete any processing. The former completely exercises the system; even though it is narrow in scope it is deep in penetration.

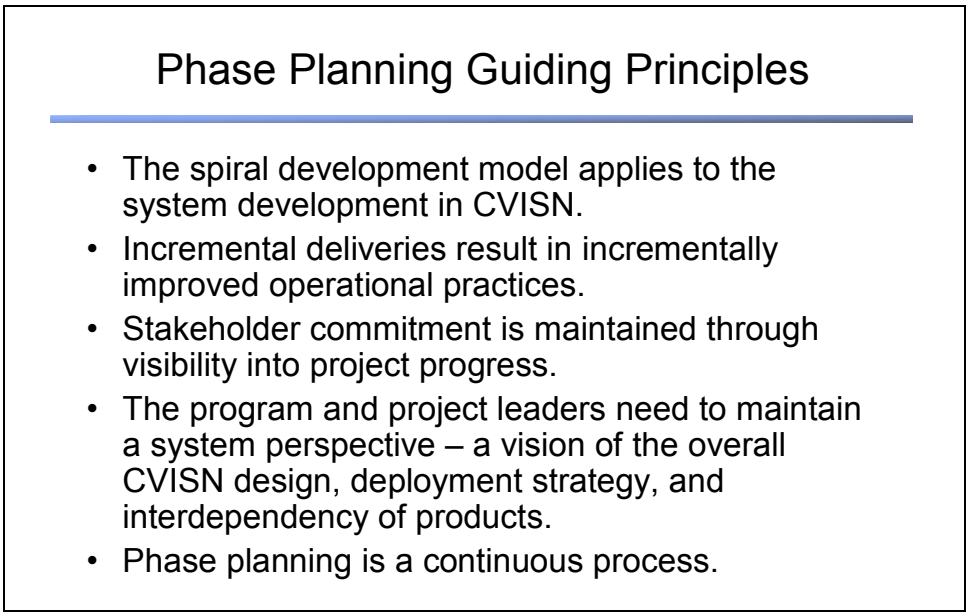


Figure 4-1. Phase Planning Principles

3. **Stakeholder commitment is maintained through visibility into project progress.** Seeing is believing (and believing is funding). Nothing succeeds like success. There are good reasons these aphorisms are well-worn – you must keep nurturing the confidence of your stakeholders as shown in Figure 4–2. Providing visibility means offering physical demonstrations of useful capability, and regular management status reporting.

Some means to accomplish this are:

- Seizing opportunities for showing off functional capability, such as a live software demo on a portable computer, or a site tour of an operational electronic screening capability at a weigh station.
- Establishing an Initial Operating Capability as early as possible so as to deliver real benefits.
- Outreach activities such as an "Industry Day" at a weigh station or truck stop.
- Regular stakeholder status reporting in the form of progress charts. Samples are included in this Guide.
- Engaging stakeholders in operational tests and assessment of results.

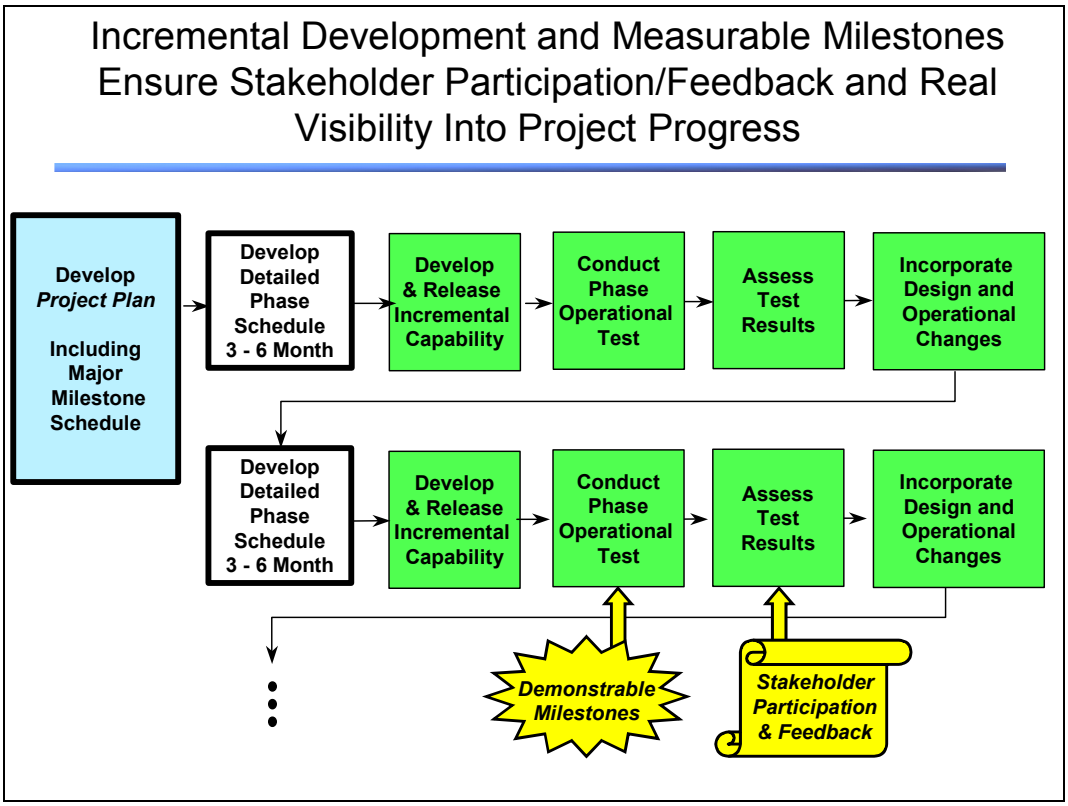


Figure 4-2. Give Stakeholders Insight into Progress

4. **The program and project leaders need to maintain a system perspective – a vision of the overall CVISN design, deployment strategy, and interdependency of products and projects.** It is very easy for the development teams "working in the trenches" to be so focused on short-term goals that they lose sight of overall efficiencies. This is a job of the System Architect, which is why that is such a key team role. This "overall vision" is more than a mere statement of objectives. It is the envisioned technical systems as captured and expressed by top-level logical design drawings, subsystem identification, interface specifications, and program-level milestones. As an example of maintaining a system perspective and adjusting it when required, over just a few years the robustness of server-hosted browser-based user interfaces with centralized database back-ends has overtaken the reasonableness of deploying stand-alone desktop computer single-user applications. On the other hand, the supposedly common sense move to the "universal" Java programming language has sometimes failed in practice, and those developers have been forced to re-tool for a better-supported but more primitive programming language environment.

5. **Phase planning is a continuous process.** Just like a marching army the project team needs to be fed. This means that costs will continually accrue – and sometimes it's easy for months to go by before you realize nothing has been accomplished. It's not plan and forget. Keep an eye on the schedule and a constant awareness of what barriers need to be removed in order to sustain progress.

4.2 Phase Planning Operational Concepts

Figure 4–3 lists the operational concepts for phase planning in CVISN.

1. **Involve the entire project team in the planning process for each phase.** You might want to schedule a phase planning mini-workshop with your team to kick off the planning for each phase. Appendix E contains some suggestions for conducting a phase planning session.

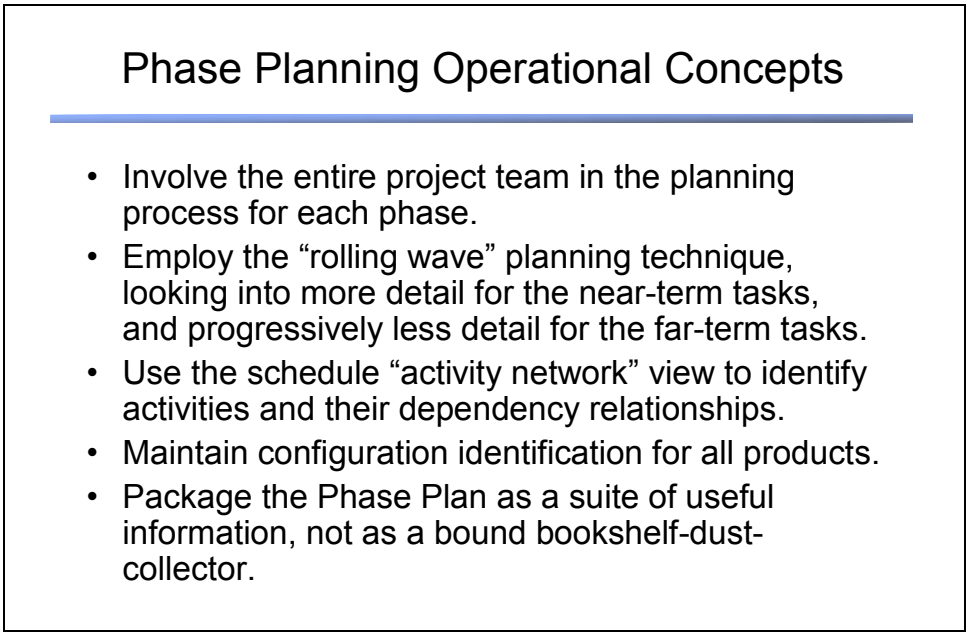


Figure 4-3. Phase Planning Operational Concepts

2. **Employ the "rolling wave" planning technique, looking into more detail for near-term tasks, and progressively less detail for far-term tasks.** For example elements at level 3 of the WBS that are currently being developed should be decomposed to level 4 or 5, whereas other elements that are being deferred until later can remain at level 3. In the detailed activity network schedule, those activities occurring in the next month or two should be broken down to an average of one person-month each, whereas activities occurring over the phase planning horizon can represent more work, say an average of six person-months each. This is portrayed graphically in Figure 4–4. You will hear the project management profession [12] call near-term activities "work packages" and far-term activities "planning packages".

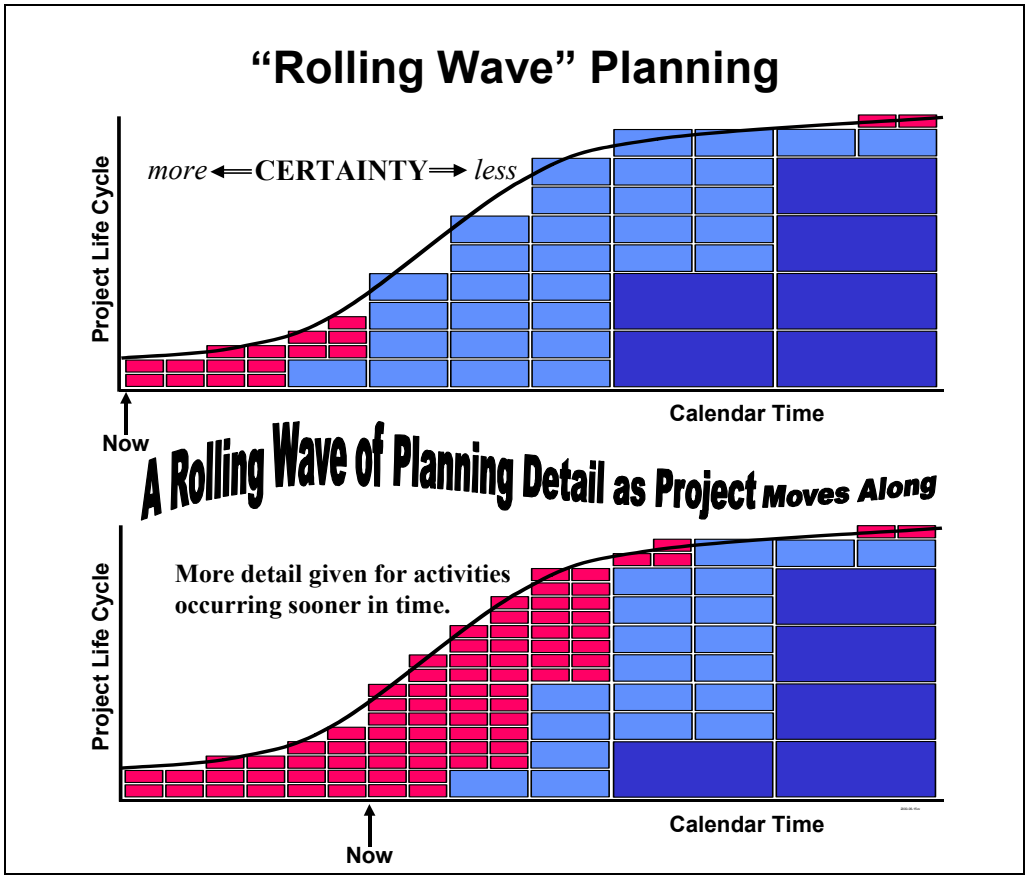


Figure 4-4. The Rolling Wave Planning Technique Decomposes Work in the Near-Term into More Detail

3. **Use the schedule "activity network" view to identify all activities and their dependency relationships.** The activity network (see Section 3.5) is a standard capability of every desktop scheduling software package. If you have been persistently attached to Gantt charts, please give the network view a try – only the network view can adequately portray the interdependence of activities. Today's desktop scheduling software tools [14] generate both the network diagram and the Gantt chart from the same activity database. As a practical matter the network view is best printed on a large-format output device such as a 36-inch roll-fed color ink jet printer. Please review Chapter 3 for more on scheduling.

4. **Maintain configuration identification for all products.** As we discussed in Chapter 2, during an incremental development and deployment approach, you will be integrating interim versions of products to achieve intermediate capabilities. Chaos will quickly set in unless your development teams can unmistakably identify the installed base of software versions and their capabilities. Remember also to keep track of the underlying commercial-off-the-shelf (COTS) products (such as operating systems, databases, communications protocols, etc.) that go along with the applications you or your vendors deploy.

5. **Package the Phase Plan as a suite of useful information, not as a bound bookshelf-dust-collector.** Unlike the program plan, which can be allowed to age, the Phase Plan must remain current and very much "alive." Therefore, it should be thought of not as a permanently bound document, but rather as a stack of pages that one might organize in a three-ring notebook and carry around to meetings. See Appendix B for ideas about keeping a Project Manager's Notebook.

4.3 Phase Planning Process Steps

Next, we describe the recommended process for planning a phase in one of your CVISN projects. This process is illustrated in Figure 4–5. You'd go through this before each new phase, say every 3 to 6 months.

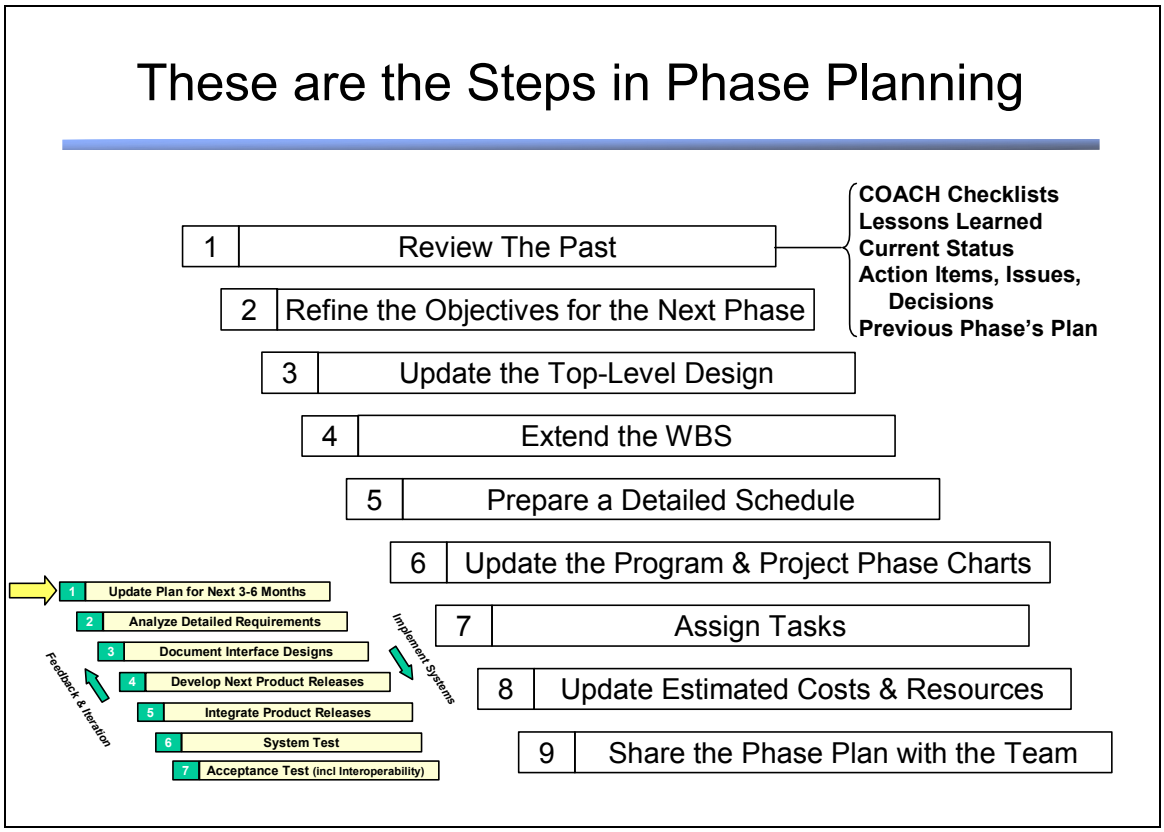


Figure 4-5. Phase Planning Process

As a reminder, please refer to Figure 1-7 for an illustration of what should be in your Phase Plan – the product of the phase planning process. Flesh-out certain elements of your project plan, each in more detail than before (see Appendix B for examples):

- Phase Objectives.
- Work Breakdown Structure decomposed to lower levels, with responsibility assigned.
- Detailed working-level schedules with tasks small enough to be assigned to individual team members (see Chapter 3 about scheduling).
- Milestones.
- Budget.
- Status charts for the previous phase, for updating to the next phase.
- Anything else you find useful.

You will be called upon to present status at a variety of meetings, and to write status reports. Designing the planning materials to make them "presentable" will simplify your life. Build your Phase Plan with materials that are easy to update, and that are readily adaptable for a presentation or as the basis for a more formal progress report.

Do as many as possible of the steps below in project team joint working sessions. That will keep team spirit up, facilitate communication, and assist new team members in getting on board. Store and maintain the planning and status materials on-line (for example, in a server directory) accessible to all the team members.

4.3.1 Review the Past (Step 1)

The project team should refresh their memories about where the project is, how it got there, and the original intentions for the future (in particular, the upcoming phase). This is illustrated in Figure 4–6. Remind the team that applying lessons learned in the past can head off problems in the future.

Review COACH Checklists
A subset of the team should quickly review all parts of the COACH (CVISN Operational and Architecture Compatibility Handbook) [15] checklists previously completed by the state's CVISN Program team. During the review, focus on the sections closely related to the work you are now planning. This review is intended as a refresher for where you are headed, and as an opportunity to update what you've said previously. As necessary, update the COACH checklists. If you are changing anything that might affect another project, be sure to collaborate with the Program Manager and that Project's Leader.

Review Lessons Learned
Think and talk about what you did in the previous phases that worked well, and what you'd like to do differently. If a process isn't working, figure out how to fix it. Give everyone on the team a chance to share and contribute, using a round-robin approach. Appendix F tells how to conduct a formal Lessons Learned session.

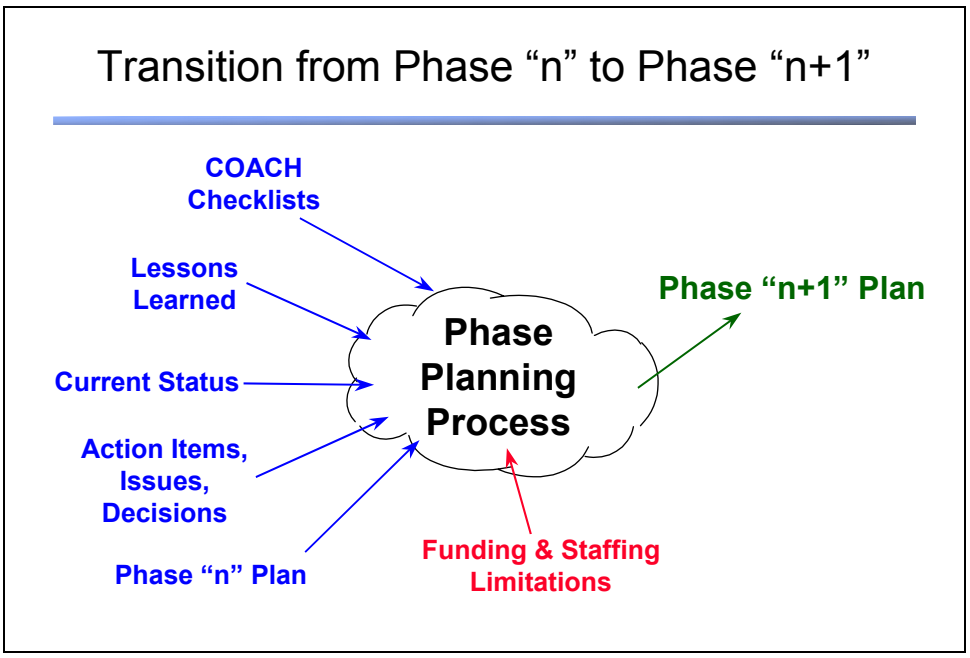


Figure 4-6. Review the Past as You Plan for Next Phase

Review Current Status

Summarize the status of key milestones on this and related projects. Review the status of external dependencies. Check planned versus actual funding and spending levels.

Review Action Items, Issues, and Decisions

This is a good time to go over the lists of program-wide and project-specific open action items and unresolved issues. Also revisit previously-made decisions. Are they still valid?

Review Previous Phase's Plan

Are there incomplete tasks to be stretched out into the next phase? Has testing been completed? Are there any repairs that need to be factored into the upcoming phase? Are there any schedule or cost lessons learned that you recognize now as you look back?

Consider the staff you have available to work on the project, and the amount of funds unspent and expected. Update the plan to reflect new constraints, the sharing of staff with other projects, etc.

4.3.2 Refine the Objectives for the Next Phase (Step 2)

When you initially defined your project plan you established high-level objectives for each project phase. Revisit those original ideas, update, and refine them. Do any unmet objectives from the previous phase carry into this one? Conversely are there any objectives originally allocated to future phases that should be accelerated and moved into this phase?

Identify which elements of the top-level design baseline are to be deployed in this phase. Then, as a team, flesh out the detailed requirements that will drive the WBS extension in Step 4. Keep the spiral model in mind!

4.3.3 Update the Top-Level Design (Step 3)

Because this is an iterative process, what your team learns during detailed development may force changes in the top-level design approach. Review and revise the top-level design as needed. Revisit the operational scenarios and functional thread diagrams; the interface specifications; the allocation of functions to systems; and the physical design. Update the design as needed.

4.3.4 Extend the WBS (Step 4)

Decompose the applicable lower-level details of the work breakdown structure. Recall from the *CVISN Guide to Program and Project Planning* [4] that a WBS is a "product-oriented hierarchy of goods and services". First, decompose products into a hierarchy of their elements; then identify the major tasks to produce them. Products to be completed in this phase should be fully-decomposed, along with integration and testing tasks, plus associated documentation, training, and field support. Minor tasks can be captured as to-do items. Don't forget to review the procurement tasks because there may be some long-lead items. WBS development is a worthwhile activity to do as a team in order to establish a common understanding, and to avoid duplication or omission within the WBS.

4.3.5 Prepare a Detailed Schedule (Step 5)

Develop a detailed schedule for the work to be accomplished during the phase. Be sure to include adequate time for technical reviews, repairs after testing, and procurement cycles.

The scheduling process is most effectively done by identifying and then linking activities per the critical path method (Section 3.5), taking advantage of a desktop scheduling software package. The output can be printed both as a Gantt (bar) chart, and as an activity network chart.

Don't forget project external dependencies, with their associated delivery or need-by dates.

4.3.6 Update the Program & Project Phase Charts (Step 6)

Recall that the *program* phase charts show what new capabilities will be provided by each *project* in a program phase. Each *project's* phase charts show what new capabilities will be provided by each *product* in a project phase. Therefore update both sets of charts once the detailed scheduling has been completed. If you had originally planned to compete an activity in this phase but you are now delaying it, check to see if other projects were depending on it, and negotiate with them.

4.3.7 Assign Tasks to Individual Team Members (Step 7)

Make sure that every task that is to be accomplished during this phase is assigned to a responsible organization and to an individual. You can record their names in the scheduling tool. Meet with those individuals to review the requirements (technical, schedule, cost constraints) and get their buy-in. Be sure their line supervisors are aware of and agree to the assignments.

4.3.8 Update Estimated Costs & Resources (Step 8)

Re-figure the estimate-to-completion for costs and resources. Ensure that you have the cash flow you need – you don't want to run out of money before receiving the next installment. Revisit the organizational staffing plan to make sure it lines up with the scheduled task assignments. Report any problems to the Program Manager, and be ready to re-work your plan.

4.3.9 Share the Phase Plan with the Team (Step 9)

Make a draft Phase Plan available quickly. Although there will still be work in some areas such as cost calculations, draft versions of everything should be available within two weeks of the team joint planning session.

Present the final Phase Plan to the project team members in a follow-up meeting of say 2 hours. This provides the forum for dissemination of details that weren't resolved in the draft planning stage. Equally important, this gives the participants a sense of "closure" by seeing tangible output from all the planning efforts.

After this follow-up session **do not worry about formally publishing it**. Let the elements of the Phase Plan remain a living set of documents to be used and revised during the phase.

Be sure to have periodic (typically monthly) review sessions to check on progress and the likely need for changes. Trust but verify – this is the subject of the next chapter.

5. CVISN PHASE TRACKING

Tracking is the measuring aspect of project management. We prefer to say "tracking" instead of the passive "monitoring" because tracking denotes an active process to seek out relevant information. The Project Management Institute (PMI) [12] defines project control as:

> *The process of comparing actual performance with planned performance, analyzing variances, evaluating possible alternatives, and taking appropriate corrective action as needed.*

Note that "performance" is always assessed relative to the plan on the **three dimensions of the triple constraint: cost, schedule, and technical performance** [18]. Fine-tuning knobs that the Project Leader can turn to assert control are [12]: scope; schedule; cost; quality; risk; and staffing. It is beyond the scope of this Guide to deal with all the various corrective actions a Project Leader might take, but they are readily found elsewhere; all project management reference texts [12, 17-21] include extensive discussions of project control strategies.

As shown in Figure 5–1, a tracking process is necessary to loop back to the plan and thereby set up a management feedback system. Process control engineers speak of "open-loop" versus "closed-loop" control systems. In an open-loop system you take aim and hope for the best. (An example from your home would be a log burning in a fireplace with the room absorbing however much or little heat it happened to produce.) In a closed-loop system the output is measured and compared to the target or set point; then corrective action is taken to re-aim the system. (An example from your home would be the furnace which has a thermostat that cycles it on and off to regulate the temperature.) Phase tracking is like the thermostat.

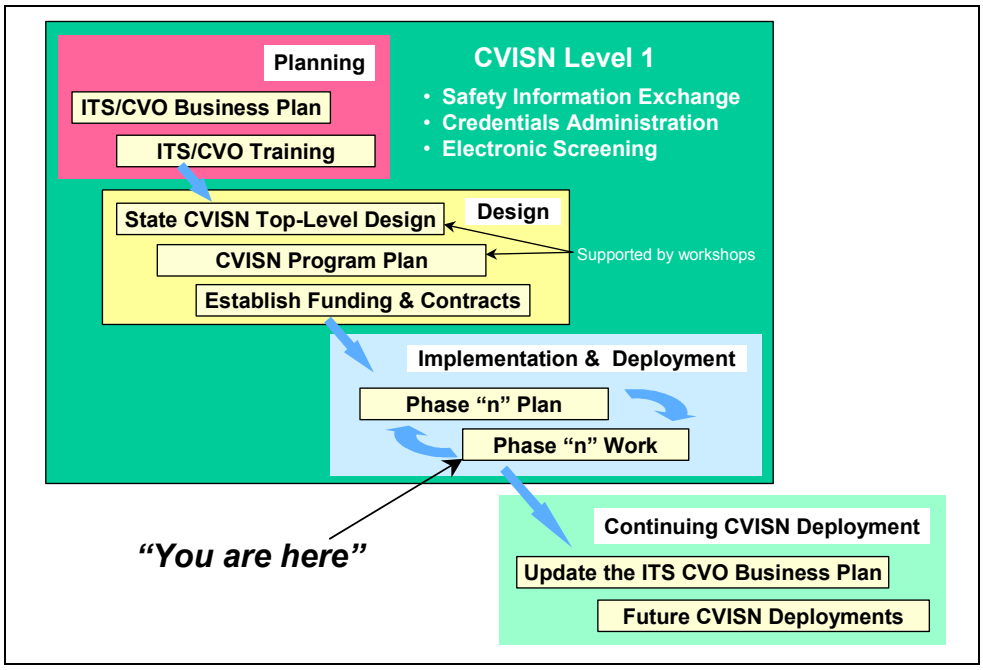


Figure 5–1. Each State Deploys CVISN Capabilities Incrementally

Pre-defining the tracking process lets each team member know what is expected of him or her. After it becomes routine, they won't complain nearly as much as when they first hear about it!

As in the previous chapter, we first discuss **guiding principles** in the sense of natural laws to follow. This is followed by a discussion of **operational concepts** that are generalized from particular instances or experience and can be applied to new situations. Next we lay out phase tracking **process steps** which you should tailor to your unique project environment and organizational culture and tools. We end this chapter with a description of the types of meetings that support phase tracking.

5.1 Phase Tracking Guiding Principles

Figure 5–2 lists the phase tracking principles for CVISN.

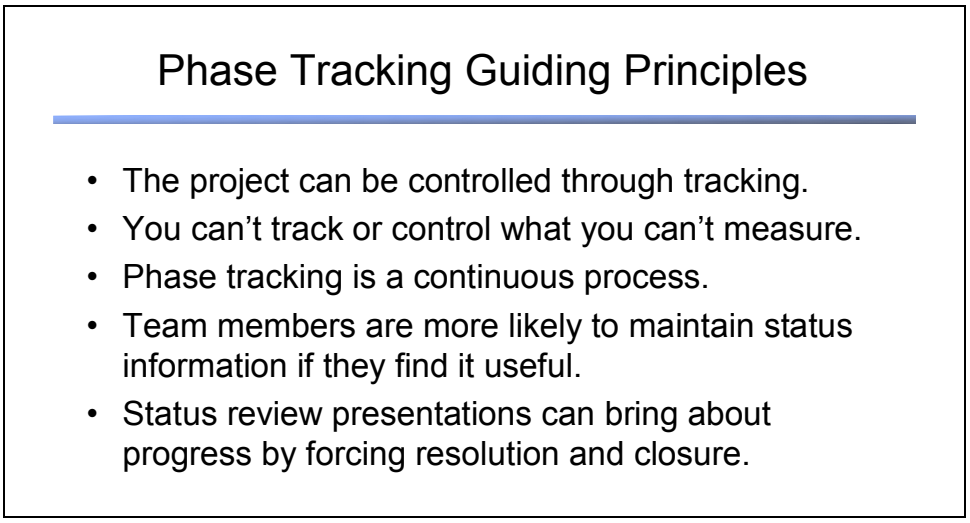


Figure 5–2. Phase Tracking Guiding Principles

1. **The project can be controlled through tracking.** A closed loop system has measuring devices that provide feedback, and a regulator that uses that information. This principle says that you should run your project as a closed loop system with management-oriented measuring devices in place, including your own instincts.

2. **You can't track or control what you can't measure.** This is why, for example, you want to enforce the use of meaningful readily-quantified milestones.

3. **Phase tracking is a continuous process.** Recall the three dimensions of the triple constraint that you must track: cost, schedule, and technical performance. Often deviations from the schedule will be your first indication that interventions are needed. That's because defining when something is supposed to be completed is often easier than estimating how much it will cost. Cost deviations will come next. Finally, technical performance will be missed. To remain cognizant of all three dimensions the Project Leader must constantly monitor various measures of progress.

4. **Team members are more likely to maintain status information if they find it useful.** In the previous chapter we talked about shaping the planning materials to suit multiple purposes. For example, if you want team leaders to keep an eye on their detailed schedules, make it easy for them to use that same scheduling tool to generate information they use for required status presentations. By choosing and formatting the elements of the Phase Plan such that they are useful in other ways, you make it more likely that those elements will be kept current.

5. **Status review presentations can bring about progress by forcing resolution and closure**. Status presentations are the hidden hand of management. If for no other reason than peer pressure and fear of embarrassment, action items get closed and technical issues resolved, often on the day before the status review but, what the heck, at least they finally get settled.

5.2 Phase Tracking Operational Concepts

Figure 5–3 lists the phase tracking operational concepts for CVISN.

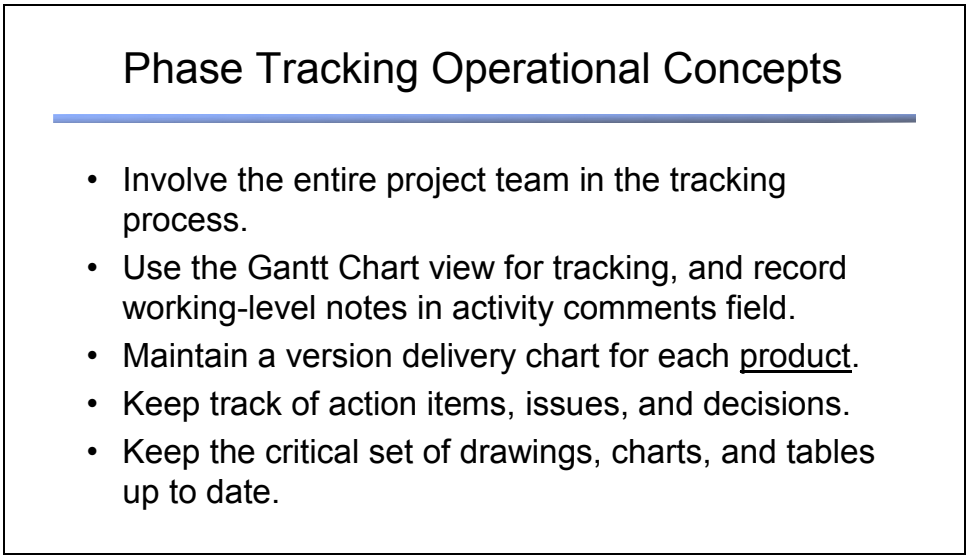


Figure 5–3. Phase Tracking Operational Concepts

1. **Involve the entire project team in the tracking process.** Everyone on the team needs to feel accountable. You might want to hold weekly status review meetings, which could include off-site personnel and contractors via teleconference. What's important is that everyone realizes that the plan does matter, and that they are supposed to work to the plan. If problems arise, be sure not to "shoot the messenger" by reacting adversely to bad news. Instead figure out how to solve the problem that was identified as a result of the tracking process.

2. **Use the scheduling tool's Gantt chart view for tracking, and record working-level notes in the tool's activity comments field.** Please review Figure 3–3 for a reminder of what a Gantt chart looks like. The Gantt chart view is automatically generated by any desktop project scheduling software package. Dates can be shown both as text in columns and as bar ends on a calendar scale. Since the activity descriptions are necessarily very cryptic, use the activity comments field for enlightening elaboration such as technical details and the working-level to-do list status.

3. **Maintain a version delivery chart for each product.** A sample version delivery chart is shown in Figure 5–4. This chart should show what capability will be provided with each

version of the product. Describe the capabilities in end-user terms, rather than in system developer terms. By annotating the chart with current status information the same chart can serve both as a vehicle for planning and for reporting. The same chart, 8 months later, is shown in Figure 5–5. Simply by relocating the highlighting arrows and updating the text, the chart shows where the development is at that later time.

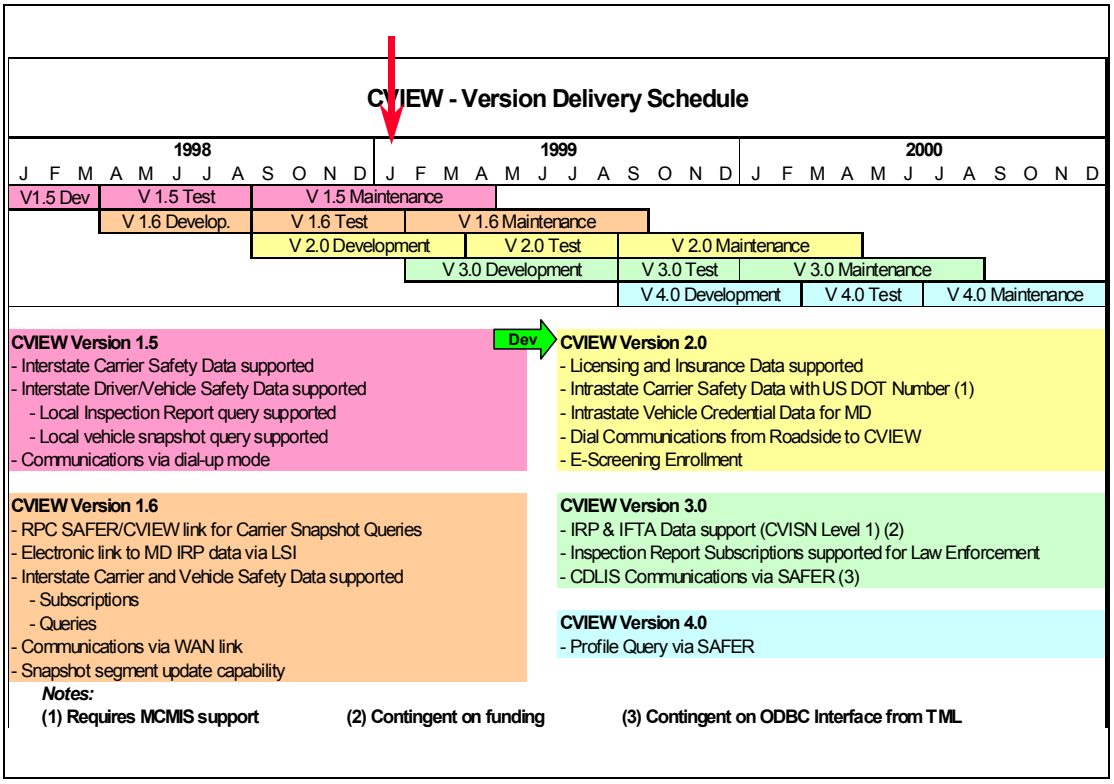


Figure 5–4. Sample Version Delivery Chart

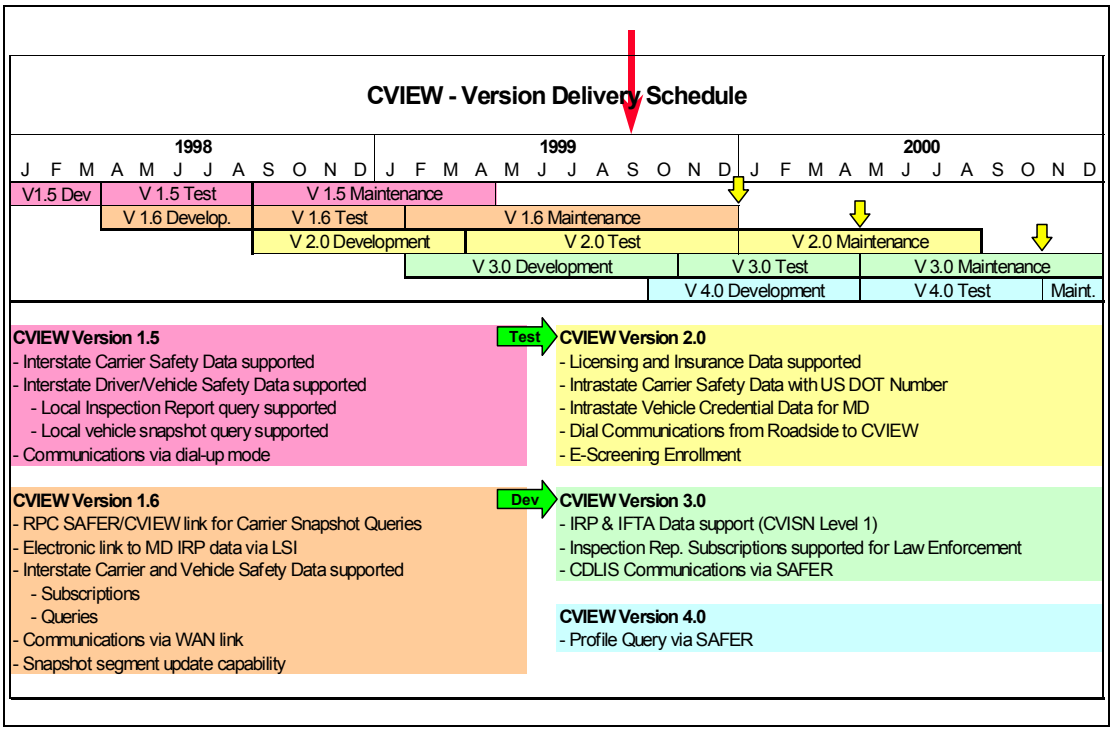


Figure 5–5. The Same Chart – 8 Months Later

4. **Keep track of action items, issues, and decisions.** These three categories capture the key programmatic information that project teams need to know. It is more economical to just record these, rather than full-blown meeting minutes. The objective is to provide structure, communication, and follow-through for topics that could otherwise easily "fall through the crack". We like to say that you assign an action item, have an issue, and make a decision. An action item is something that one person or a small group will work on and resolve in a limited period of time. "Resolving" may only mean gathering information so that the larger team can make a decision. An issue is something one is worried about but can't define well enough to assign as an action item. A decision is a conclusion agreed to by the team. There is a natural flow from a hazy issue or concern, to an action item, to a decision, as shown in Figure 5–6. Your project could record action items, issues, and decisions simply by creating a paper form that staff members fill out and turn in to a master list-keeper. Appendix G has a sample form to use. A more sophisticated implementation would, for example, maintain such lists in a Web-enabled database.

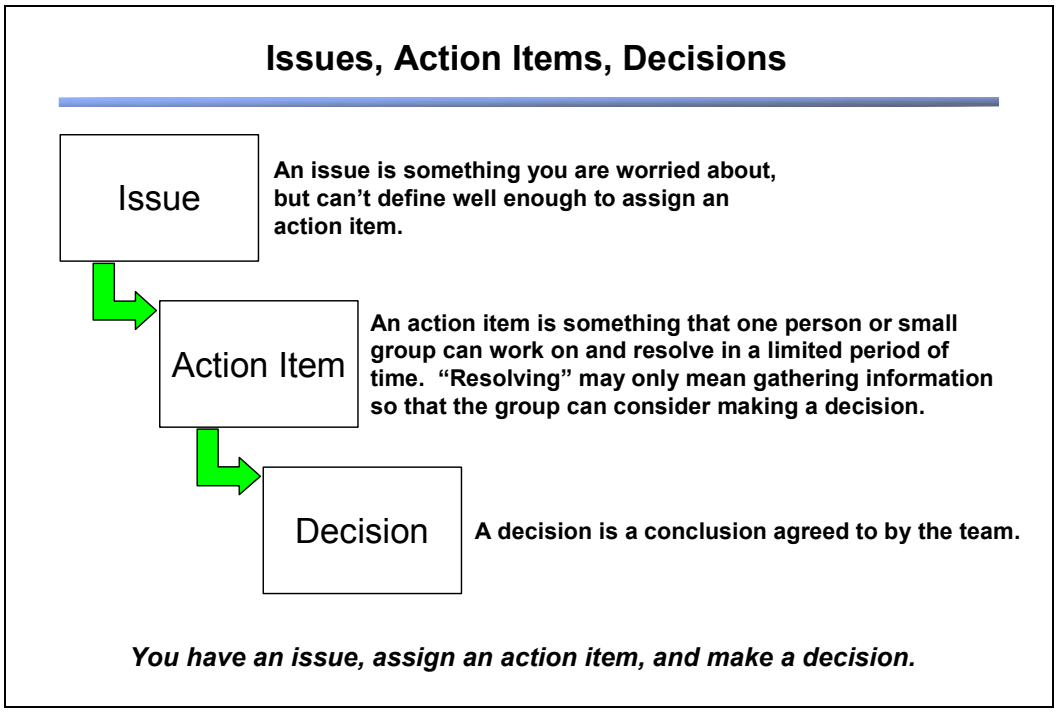


Figure 5–6. Flow of Issue to Action Item to Decision

5. **Keep the critical set of drawings, charts, and tables up to date.** Make them part of your Project Manager's notebook and carry them with you. Spread them out on walls and tables during status meetings. Mark them up. Use them to remind you of the big picture, even when you are buried in detailed problem-solving. For example, sketch network connection modifications not in thin air during a meeting, but instead by marking up the physical design diagram placed on the conference table.

5.3 Phase Tracking Process Steps

Any task involving more than one person, or lasting more than one month, needs to be tracked. Earlier in this chapter we talked about closed-loop versus open-loop systems. Figure 5–7 illustrates that tracking mechanisms are used to "close the loop" in your project management system.

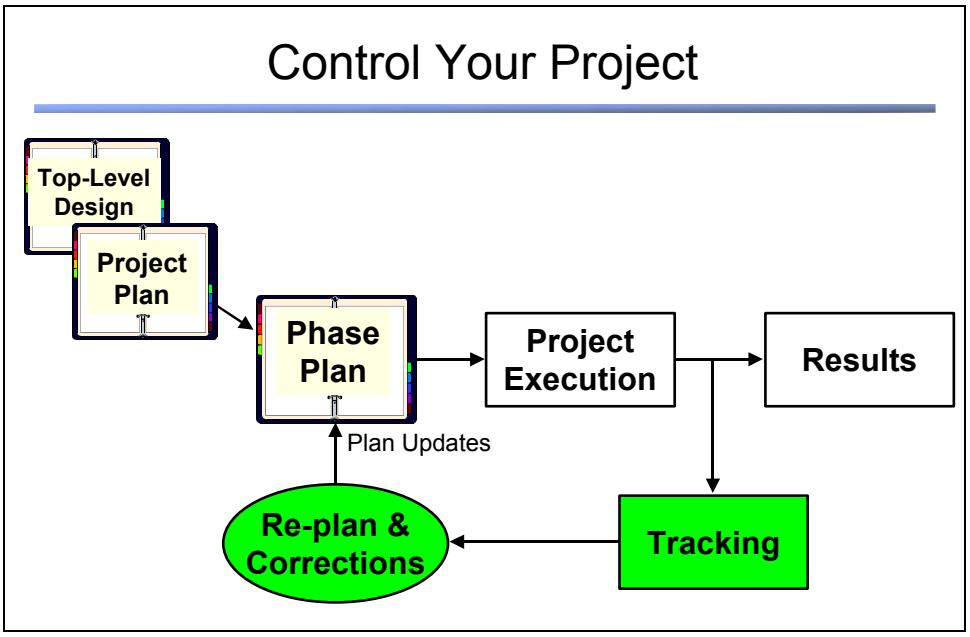


Figure 5–7. Tracking Can "Close the Loop" in Your Project Management System

Tracking should be a continuous process. Below we describe several practical and proven process steps for project tracking as illustrated in Figure 5–8. All the methods involve people communicating with each other, either directly or through written materials.

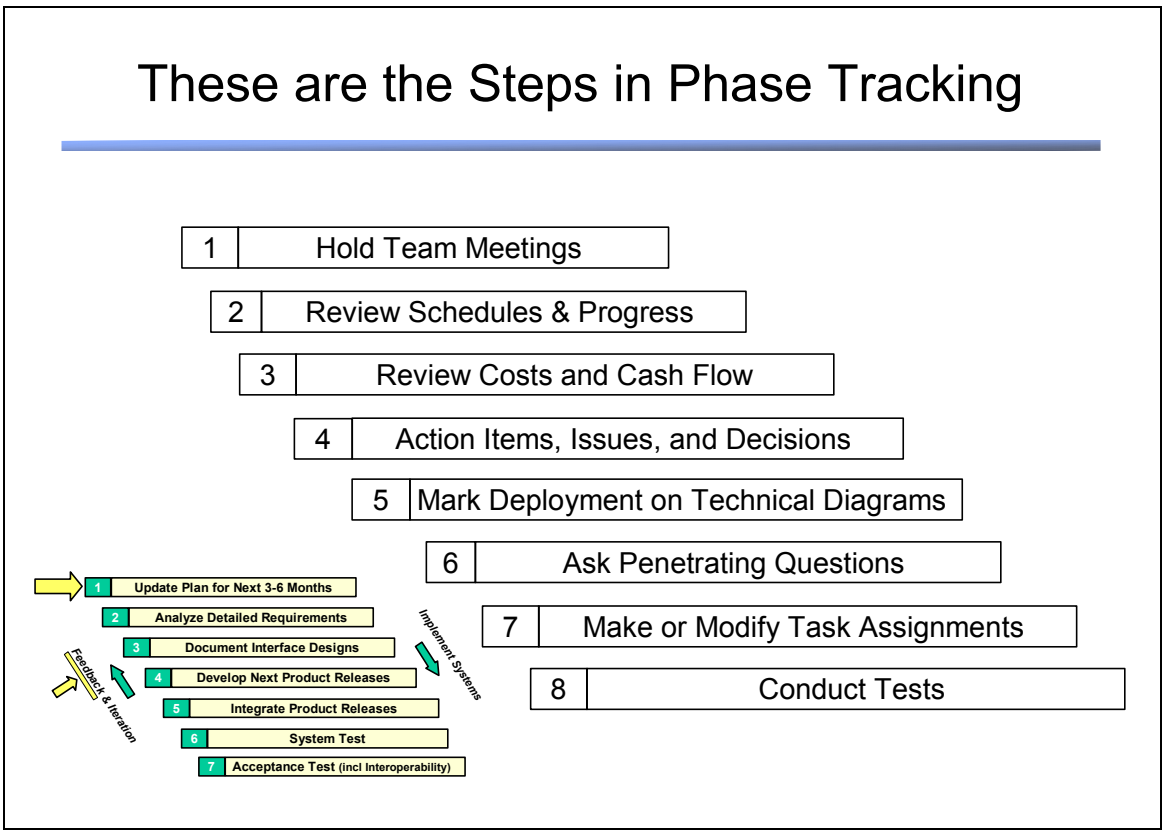


Figure 5–8. Phase Tracking Process

Remember to track in all three areas of the **triple constraint: cost, schedule, and technical performance**.

Cost reviews are usually most effective if accomplished in the smallest possible group, often one-on-one. In addition to staffing costs look ahead for procurement, travel, and other lump expenses. Beware of delayed lump expenses that haven't hit the books yet.

Schedule reviews usually involve the team. What one task is experiencing may affect other tasks. Often schedule slips can be avoided by applying different resources to a problem. Reassignment of priorities or tasking is best accomplished with the concurrence of the whole team.

Technical performance is usually assessed through testing. Different kinds of tests are performed by different teams, and at different points in the development process. In each phase, some kind of testing should be performed to demonstrate that the phase objectives were met. Analysis of test results should be completed promptly to assess the success or failure of the testing process. Factor in the calendar time that is inevitably required for repairs and rework to address the problems found during testing. Please see the *CVISN Guide to Integration and Test* [6] for a complete discussion of test planning and conduct.

The next several sections discuss ways of assessing cost, schedule, and technical performance.

5.3.1 Hold Team Meetings (Step 1)

People sometimes complain about "time wasted in meetings" but surely even more time would be wasted if there were never team meetings, due to the lack of communication and coordination that would arise. You need to seek the right balance for the frequency and duration of meetings. Different kinds of meetings accomplish different objectives; all require an agenda in advance. The kinds of meetings that are typically useful are summarized below, and discussed in more detail in Section 5.4:

- Weekly project team meetings are held to discuss progress and problems with product development and integration. Schedules are reviewed. Special "issues" meetings are called as needed to address a particularly thorny topic, often involving only a subset of the team.
- Monthly program team meetings are held to review the status of each project. Schedule changes for intermediate deliveries from one project to another within the program are identified and discussed at program team meetings. Separate meetings are held as needed to discuss issues and/or make decisions about areas that affect more than one project.
- Quarterly state meetings are held to exchange ideas and status information with executive management and advisory groups.
- Test review meetings are held to discuss plans for tests, and to discuss test results.

Not every meeting needs to be face-to-face – some can be accomplished via telephone. Long-distance carriers offer teleconference bridge services: everyone calls in to one number and can then talk to and hear everyone else. Participants can also share computer terminals for real time review and modification of documents. Software packages such as Microsoft® NetMeeting make this relatively easy over the Internet. Security modifications are typically required – your network firewall administrator will have to open up certain ports. You might want to purchase a headset so that you can be on the phone with your hands free for the computer terminal.

5.3.2 Review Schedules and Progress (Step 2)

At the program level, you set phase objectives and milestones that are usually related to project integration. At the project level the objectives are limited to the scope of the project, and the milestones are usually related to product integration. At the product level, version delivery schedules are set. Review whichever schedule is at the appropriate level during your team meetings. Give ample time to address areas where progress is slower than planned. Help, don't punish, areas that are behind schedule. Reward openness. All of this will try your patience! Keep a sense of humor and perspective.

It is customary to ask about percent completion on tasks. Reference [17] advises *"Don't ask [first] what percent complete their current task is. This is a very ineffective question in software and technology projects. Instead, ask them what is left to do in the task."* Only after that, assess completion on a percentage basis.

5.3.3 Review Costs and Cash Flow (Step 3)

It is customary to plot three curves on one report: planned expenditures, actual expenditures, and funding. Beware that comparing actual expenditures to planned expenditures is adequate only if nothing is behind schedule. The monthly expenditure "burn rate" is relatively stable; if the team is behind schedule then monthly costs will continue to accumulate over the additional time periods required to finish – and you will end up over budget at the end.

5.3.4 Review and Maintain Lists of Action Items, Issues, and Decisions (Step 4)

In all team meetings, you should make written action item assignments as needed, and then follow up on them. By convention action items need "formal" closure via memo or e-mail. It's especially important to reach closure soon for action items that affect the current phase. Decisions that affect the delivered baseline should be handled through your configuration management processes. When new issues arise, note them. If old issues disappear, note that too. As an issue matures it typically transforms into an action item. When decisions are made, make a note and be sure the team is aware of the decision. Some decisions may be later rescinded or overtaken by event. Periodically review these lists to keep them tidy and current.

5.3.5 Mark Deployment on Technical Diagrams (Step 5)

Make it obvious which elements of the design baseline have been deployed. Update diagrams accordingly, for example by color-coding those items already functioning. Check off the phase deliverables. Utilizing staff-produced diagrams as part of the tracking process makes it meaningful for both staff and for management.

5.3.6 Ask Penetrating Questions (Step 6)

Ask team members open-ended questions like "What is your next step?", or "After you get that data how will you analyze it?", or "What if that approach doesn't work?" Ask even the seemingly obvious such as "When you get it, do you know where you will put it?", or "Did you actually open the box and count the units, because the last time we ordered two and only received one!"

Rosenau [18] asserts that if you don't ask questions some people won't volunteer critical information. He suggests asking (in a non-threatening manner) these particular non-directive questions:

- *What is your greatest concern?*
- *Do you anticipate any problems that we haven't talked about yet?*
- *What persistent problems do you have, and what is being done to correct them?*
- *Do you need any resources you do not yet have?*
- *Do you know of anything that will give you schedule difficulties?*
- *What kind of help would increase your confidence in the schedule?*
- *Is there anything I can do to help?*

5.3.7 Make or Modify Task Assignments (Step 7)

Unassigned tasks need to be assigned, and under-performing tasks may need to be reassigned. Staff will continue to need coaching and personal contact that can best be handled via "management by walking around". Keep an eye on the availability of personnel. Individuals may be juggling tasks on multiple projects. Have backup personnel available to carry on tasks during short-term absences such as vacation; this not only keeps the project on track but also provides valuable cross-training and skill-building for the backup personnel. Work with line supervisors to resolve problems. For your contractors this requires a degree of formality: at the minimum, written instruction via e-mail or fax. Be sure staff receive training when they are asked to tackle new technical areas.

5.3.8 Conduct Tests (Step 8)

Genuine product testing demonstrates whether or not individual products and integrated products perform according to user needs, within design parameters, and toward program objectives. Developers usually perform product-level performance testing. End-users usually perform project-level acceptance testing. Program-level testing is often performed by independent test and evaluation organizations.

Please consult the *CVISN Guide to Integration and Test* [6] for a full discussion of principles, techniques, and tips about integrating and testing components. Be sure to document test results; often enough problems supposedly fixed earlier reappear in subsequent releases. A test results package should include:

- Test input and output (archived data files, printouts, screen captures, etc.)

- Test procedures
- Problems found
- Corrective action planned

Ensure there is sufficient information captured as part of a test results package such that those tests could be re-run months later (called regression testing) to make sure that a formerly-existing capability hasn't been clobbered.

5.4 Meetings to Support Phase Tracking

The next several subsections discuss in more detail the different types of meetings mentioned earlier. They are opportunities to apply the phase tracking processes described in Section 5.3.

5.4.1 Weekly Project Team Meetings

A weekly review of the activity schedule is one of the most powerful tools available to a Project Leader. As part of the weekly team meeting, you should:

- Mark up the detailed schedule to capture the work getting accomplished. Either one-on-one, or as a team, update percent completion on activities and add detailed notes within each activity's comments field.
- Keep both the activity network (see Figure 3-2) and Gantt chart (see Figure 3-3) in front of you.
- Be prepared to dynamically alter priorities and assignments to work around problems.
- Review each product's version chart (see Figure 5–4) as a reminder of the capabilities scheduled for the current version.
- Watch out for slips in the schedule that will affect other projects and, conversely, slips in other projects that will affect yours.
- Review technical drawings such as thread diagrams and note what functions and system elements have been deployed (see Figure 5–9). Review the physical design and indicate what components now communicate, or to highlight problem areas (see Figure 5–10).
- Check on the status of open action items and configuration change requests.
- Listen for new issues, and for progress on old ones.
- Make sure task assignments are clear. Identify where new resources are needed, or where someone is underutilized.
- Discuss procurement items. Find out if paperwork is stuck in the approval chain, and who is working to break it loose.
- Ask questions. Encourage conversations. Especially make sure you give everyone a chance to talk.

Make no mistake – this is painstaking, detailed work. But the payoffs are significant, real, and immediate.

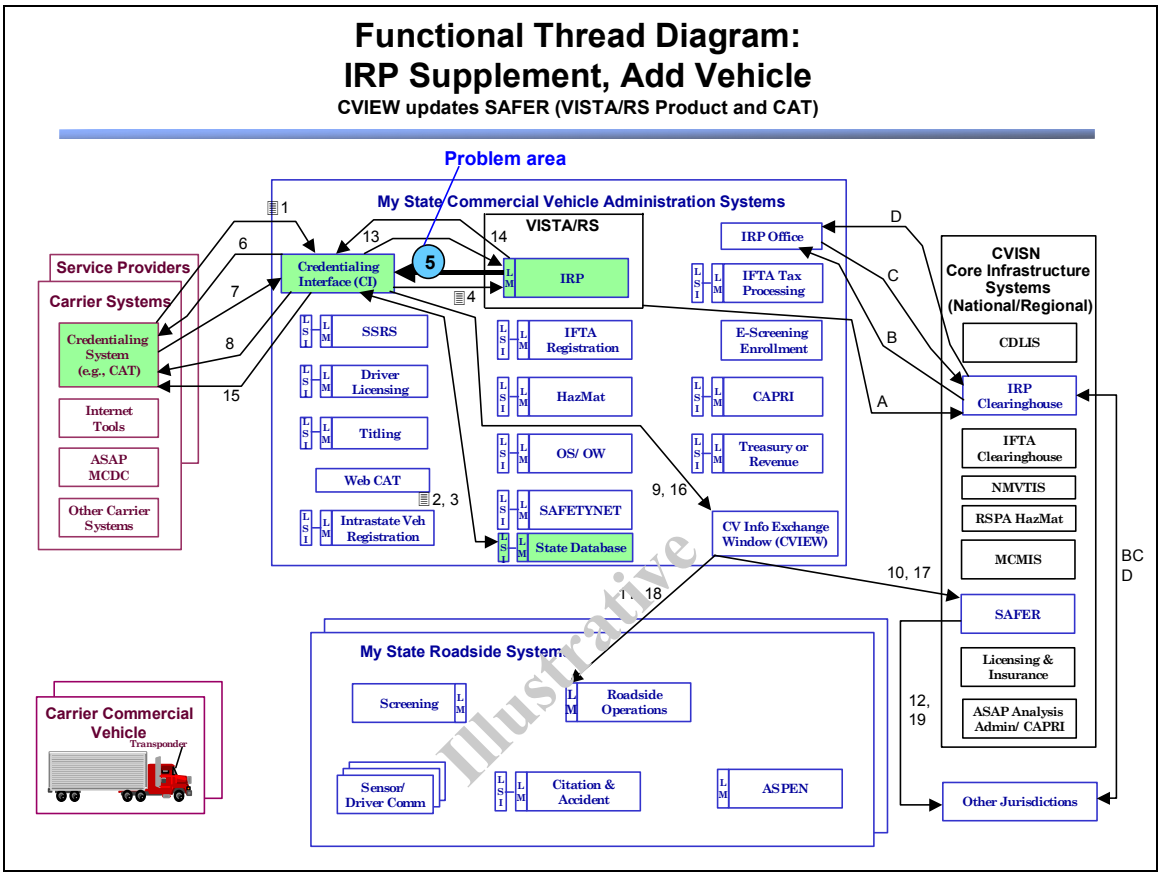


Figure 5–9. Use Your Thread Diagrams to Mark Progress and Problems

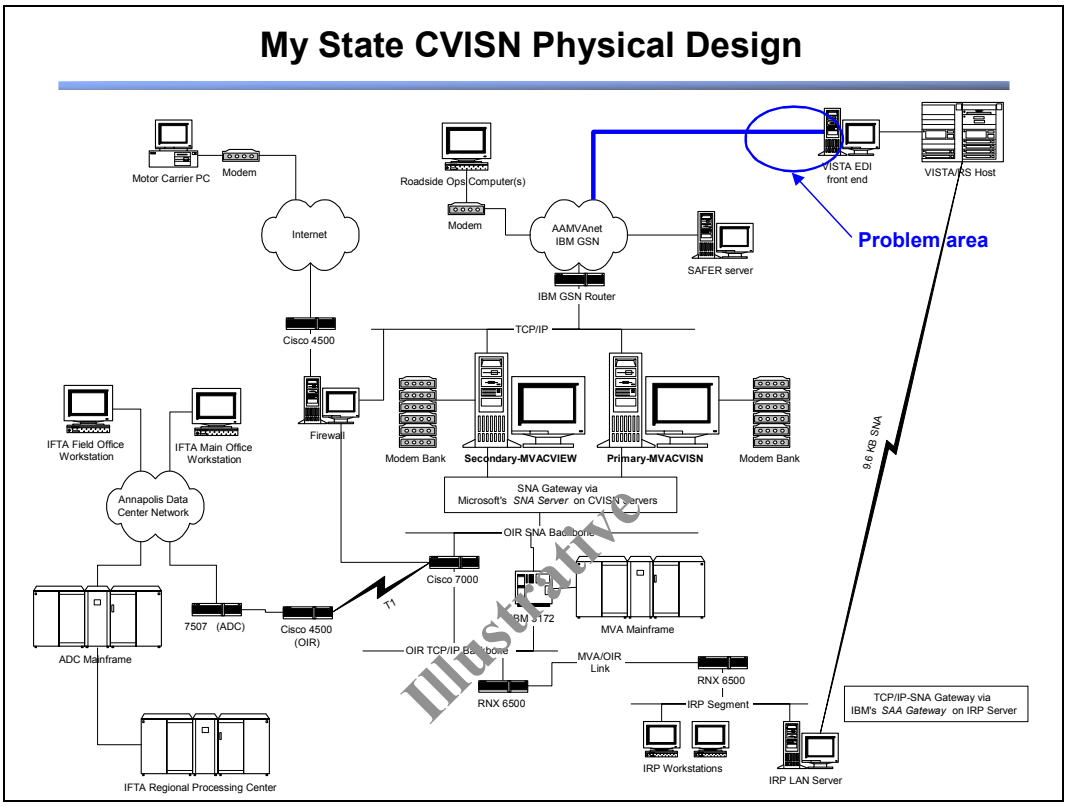


Figure 5–10. Use Your Physical Diagrams To Visualize Where Problems Are

Figure 5–11 is an illustrative photo taken at a Maryland schedule review meeting, showing the tools placed out on the conference table to assess progress and technical issues for the Maryland CVISN deployment.

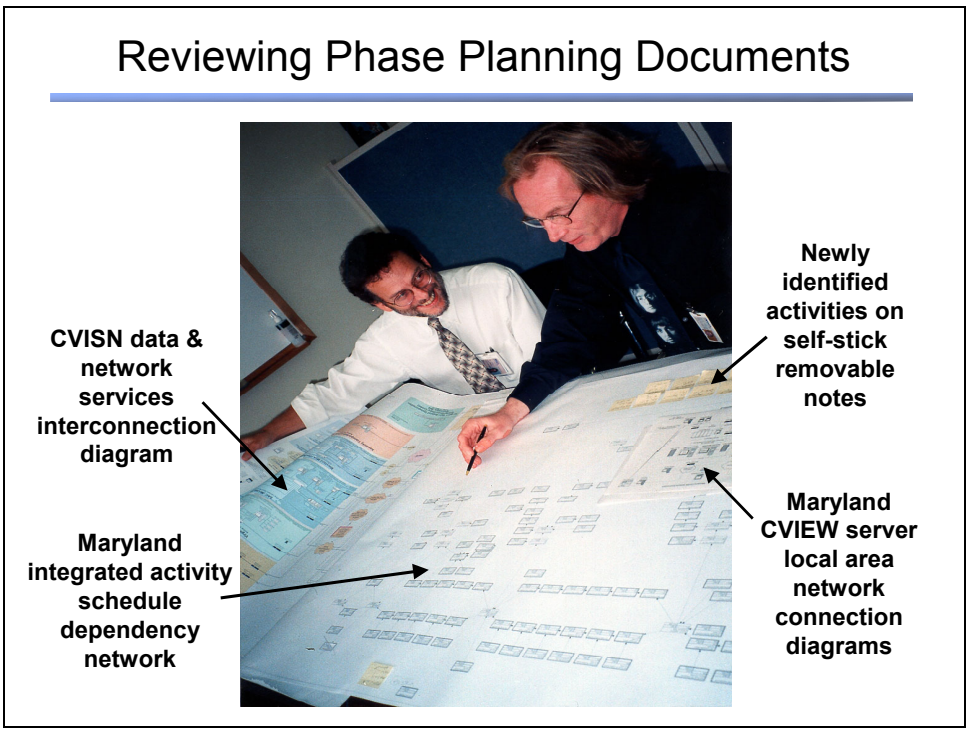


Figure 5–11. Maryland System Architect and Director of Computing Services Meet Weekly To Track Progress

5.4.2 Monthly Program Status Meetings

Monthly somewhat-formal status presentations will be informative to management, and perhaps more importantly have a magic way of coalescing issues and forging progress. Ask each Project Leader to make a 5–15 minute presentation on the status of their project. Get into a routine of using standard templates and updating them from month-to-month. This is a helpful discipline for the program team, and even a non-technical audience (e.g., a steering or oversight committee) soon becomes expert at reading and understanding the status reports. A good set of material for each Project Leader to present is:

- Project News Bulletins
- Project Phase Charts (focused on the current phase)
- Major Milestone Schedule
- Product Version Charts (optional)
- Accomplishments & Highlights Last Month
- Key Objectives & Activities Next Month
- Issues
- Design Diagrams to Illustrate Key Points
- Work Breakdown Structure annotated with cost, schedule, and technical status

Appendix C is an example of a recommended presentation format for use by a single project to make a presentation at a regular monthly program status review.

5.4.3 Quarterly State-Level Meetings

Quarterly fairly-formal presentations to executive management and steering committee members will be informative and also serve to reinforce commitment to the program from these highest levels. At such a quarterly meeting, the CVISN Program Manager (and perhaps the System Architect) makes an approximately 30 minute presentation on the status of the CVISN program. Sometimes, nearby states present status to each other as a way of benefiting from lessons learned, and keeping track of progress on program elements that are shared. A good set of material for each Program Manager to present is:

- Program News Bulletins
- Program Phase Charts (focused on the current phase)
- For each project:
    - Accomplishments & Highlights Last Quarter
    - Key Objectives & Activities Next Quarter
    - Issues
- Design Diagrams to Illustrate Key Points
- Work Breakdown Structure annotated red/yellow/green
- Capability summary

Appendix D is an example of a recommended presentation format for a quarterly program status report.

5.4.4 Test Review Meetings

Test review meetings are held initially to discuss plans for tests, and later to review test results. Participants include the developers of the components under test, the test conductors, and the analysts.

In a test plan review meeting, the focus is on what is to be tested, the configuration of the items under test, when the tests will be run, who is responsible for each aspect of testing, and how the tests will be accomplished. Test plans, procedures, data, and tools are examined.

In a test results review meeting, the focus is on the test after-the-fact. Test reports or real-time results are reviewed, problems identified, and remedy plans made. Remember that testing is intended to find errors. Be happy when errors are discovered during testing because the alternative is that they won't be discovered until the system is being used operationally.

5.4.5 Issues Meetings

Sometimes special "issues" meetings are called as needed to address a particularly thorny topic, often involving only a subset of the team. For an issues meeting to be effective you should set clear objectives, define what you need from each participant in advance, and lay out an agenda. Allot adequate time to each aspect of the issue, and make sure the conversation doesn't stray too far off track. Establish an atmosphere of open, non-judgmental exchange, so that everyone feels that they are being heard, and that their inputs are valued. Look beyond symptoms to root causes; ask "why" three times. Assign action items with due dates, and record decisions.

For all meetings, keep track of the discussions, and distribute minutes as necessary.

APPENDIX A. REFERENCES

1. ***CVISN Website*** hosted by the Johns Hopkins University Applied Physics Laboratory at http://www.jhuapl.edu/cvisn

2. ***CVISN Toolkit CD ROM.*** Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. Available at [1]. *A comprehensive set of technical documentation and planning tools assembled on a CD-ROM to assist new CVISN deployment states in the development of their CVISN Project Plans before, during, and after the CVISN Workshops. Contains a subset of the documents available on the website [1].*

3. ***Introductory Guide to CVISN.*** Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7186. Available at [1].

4. ***CVISN Guide to Program and Project Planning***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7188. Available at [1]. *Appendix A lists valuable project management references, including quite a few that published "lessons learned" from ITS deployments throughout the country.*

5. ***CVISN Guide to Top-Level Design***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7187. Available at [1].

6. ***CVISN Guide to Integration and Test.*** Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7194. Available at [1].

7. ***CVISN Guide to Safety Information Exchange***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7191. Available at [1].

8. ***CVISN Guide to Credentials Administration***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7192. Available at [1].

9. ***CVISN Guide to Electronic Screening***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7193. Available at [1].

10. ***The Art of Systems Architecting***, by Eberhardt Rechtin and Mark W. Maier. CRC Press, 1996. ISBN: 0849378362. *A must-read for the system architect; the heuristics are gems. Waterfall versus spiral development models are discussed beginning on page 93 in that document.*

11. ***Software Engineering -- A Practitioner's Approach***, 5th edition, by Roger S. Pressman. McGraw Hill, Inc., 2001. ISBN 0073655783. *Waterfall versus spiral development models are discussed beginning on page 26.*

12. ***Guide to the Project Management Body of Knowledge,*** published by the Project Management Institute [13]. Available free at http://www.pmi.org/publictn/pmboktoc.htm *Your first line of defense for terminology and concepts; concise and precise. PMI is a reputable and authoritative organization.*

13. **Project Management Institute (PMI)**
Four Campus Boulevard
Newtown Square, PA 19073
610-356-4600
http://www.pmi.org
PMI conducts monthly meetings throughout the U.S. Call to find the local chapter near you.

14. ***Desktop Automated Scheduling Tools (often called Project Management Software):***
- Open Plan from Welcom Corp of Houston, TX
- Project Planner from Primavera Systems Inc. of Bala Cynwyd, PA
- Project Scheduler from Scitor Corp. of Menlo Park, CA
- Project Workbench from ABT Corp of New York, NY
- Dekker Trakker from Dekker Ltd of San Bernardino, CA
- Microsoft Project from Microsoft Corp of Redmond, WA

15. ***CVISN Operational and Architectural Compatibility Handbook (COACH).*** Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. Available at [1].
Part 1 - Operational Concept and Top-Level Design Checklists, POR-97-7067
Part 2 - Project Management Checklists, POR-97-7067
Part 3 - Detailed System Checklists, POR-97-7067
Part 4 - Interface Specification, POR-97-7067
Part 5 - Interoperability Test Criteria, POR-98-7126

16. ***Software Reviews and Audits Handbook***, by Charles Pl. Hollicker. John Wiley & Sons, Inc., 1990. ISBN 0-471-51401-2. *60% of the pages are appendices with checklists and templates for every imaginable review.*

17. ***Breakthrough Technology Project Management***, by Bennet P. Lientz and Kathryn P. Rea. Academic Press, 1999. ISBN 0-12-449970-8. *List of reasons why systems projects fail p12; why they succeed p16. The authors argue that project tracking should be issues oriented not status oriented. One quarter of the book is devoted to addressing 56 specific common project issues.*

18. ***Successful Project Management***, third edition, by Milton D. Rosenau, Jr. John Wiley & Sons, Inc., 1998. ISBN 0-471-29304-0. *Emphasizes working within the "triple constraint" of cost, schedule, and technical performance. Has 5 chapters on monitoring progress.*

19. ***The Complete Idiot's Guide to Project Management***, by Sunny and Kim Baker. Alpha Books, 1998. ISBN 0-02-861745-2. *Has 5 chapters in the section on "The Controlling Phase", including how to solve the most predictable project problems before they start.*

20. ***Essentials of Project and Systems Engineering Management***, by Howard Eisner. John Wiley & Sons, Inc., 1997. ISBN 0-471-14846-6. *Project situation analysis is discussed on page 87 of that document.*

21. ***Project Management – A Systems Approach to Planning, Scheduling, and Controlling,*** sixth edition, by Harold Kerzner, Ph.D. John Wiley & Sons, Inc., 1998. ISBN 0-471-28835-7. *A mature handbook, broad in scope, for every aspect of project and organizational management; topics are discussed in depth. Includes a chapter on trade-off analysis, and a chapter on risk management.*

22. ***The Team Handbook***, Second Edition, by Peter R. Scholtes, Brian L. Joiner, and Barbara J. Streibel, Joiner Associates Inc., 1996. ISBN 1-884731-11-2. *Sound advice about running a team-oriented meeting, such as soliciting comments in a round-robin manner, and multi-voting to refine and select alternatives.*

23. ***IEEE Standard Glossary of Software Engineering Terminology***, IEEE Std 610.12. Published by the Institute of Electrical and Electronics Engineers, Inc. Available for a fee at http://standards.ieee.org/catalog.

24. ***ITS/CVO CVISN Glossary***. Prepared for FMCSA by the Johns Hopkins University Applied Physics Laboratory. POR-99-7188. Available at [1].

This Page Intentionally Blank

APPENDIX B. PROJECT LEADER'S NOTEBOOK

This Appendix describes a method for keeping up-to-date key information readily at hand as shown in Figure B-1.

B.1 Tab 1 – Red-Lined Project Plan

Some key pages of the published Project Plan are handy to have close by, such as the latest:

- Project Organization Chart
- Contact list (names and telephone numbers)

The rule of thumb is to keep up-to-date those items that people ask for.

Make red-line changes to assist you if and when you need to formally republish the Project Plan, but don't feel constrained by the "need" to republish it periodically.

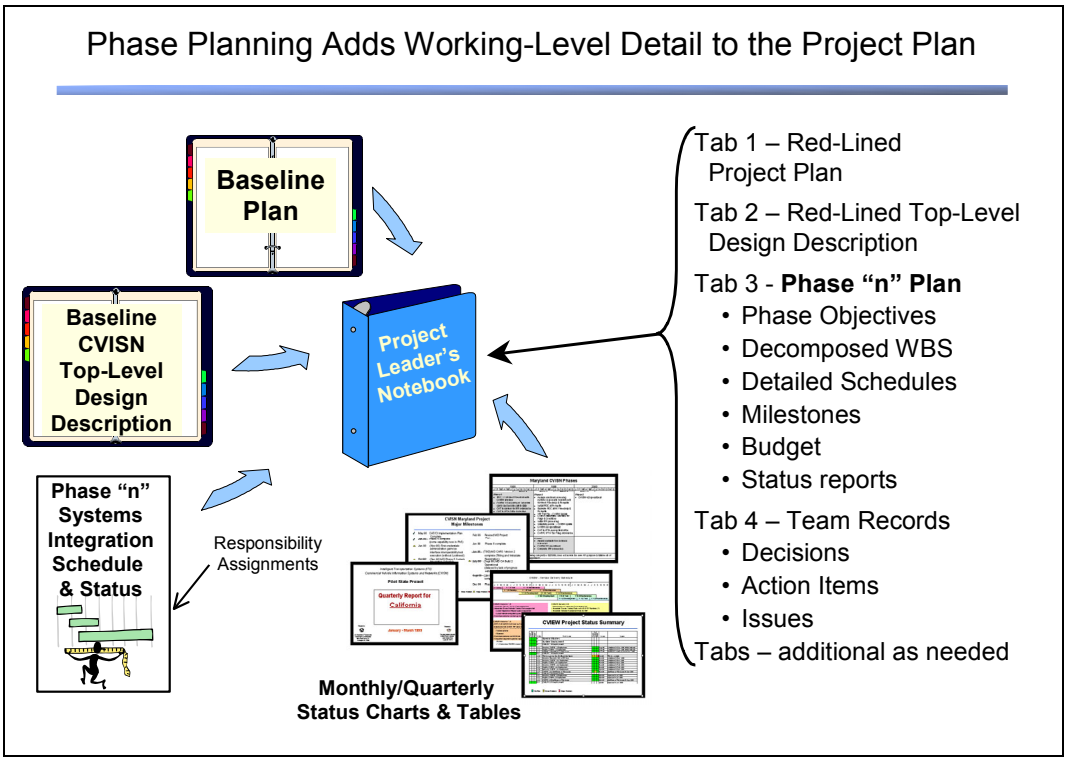


Figure B-1. Project Leader's Notebook Keeps Up-to-Date Phase Plan Readily at Hand

B.2 Tab 2 – Red-Lined Top-Level Design Description

As the development proceeds, requirements evolve; computers with expanded capability become available; new versions of commercial off the shelf (COTS) products (e.g., operating systems, databases, spreadsheets, browsers) are released; and end-users gain experience working with your prior deployments. All of these might be grounds to update the baselined top-level requirements and design. Manage such changes using your configuration management processes. Keep the design red-lined so that you have ready access to current drawings, tables, checklists, etc.

B.3 Tab 3 – Phase Plan

Visualize the Phase Plan as a suite of readily-useful key planning and tracking charts as opposed to a wordy bound document. Imagine a mountain climber – his or her base camp is equivalent to our Project Plan, but their backpack is equivalent to our Phase Plan. Weight counts; so carry only the necessary tools: the phase planning gear necessary to tackle the work immediately ahead.

Figure B-1 illustrates the essential elements of the Phase Plan, packaged into the Project Leader's Notebook. The Project Leader can write some of the elements, but others must come from the individual development team leaders.

If there are issues that cannot be decided while preparing the Phase Plan, simply document them as such and move on. Do not lose the issues, but do not get bogged down.

B.3.1. Phase Objectives

The objectives for each phase should be clearly stated so that all stakeholders can understand them. Often these are simply a bulleted list of target capabilities such as that shown in Figure B-2, which is an example of a program-level phase chart. A project-level or product-level phase chart would summarize the target capabilities planned for just one project or product in the upcoming phases.

B.3.2. Decomposed WBS, with Tasks Assigned

When you completed your Program or Project Plan, you developed the upper levels of the work breakdown structure (WBS). The details portrayed in the lower levels of the WBS have been evolving as the program/projects proceeds. Update the WBS to fill in the missing detail for the work to be done as you come to understand it.

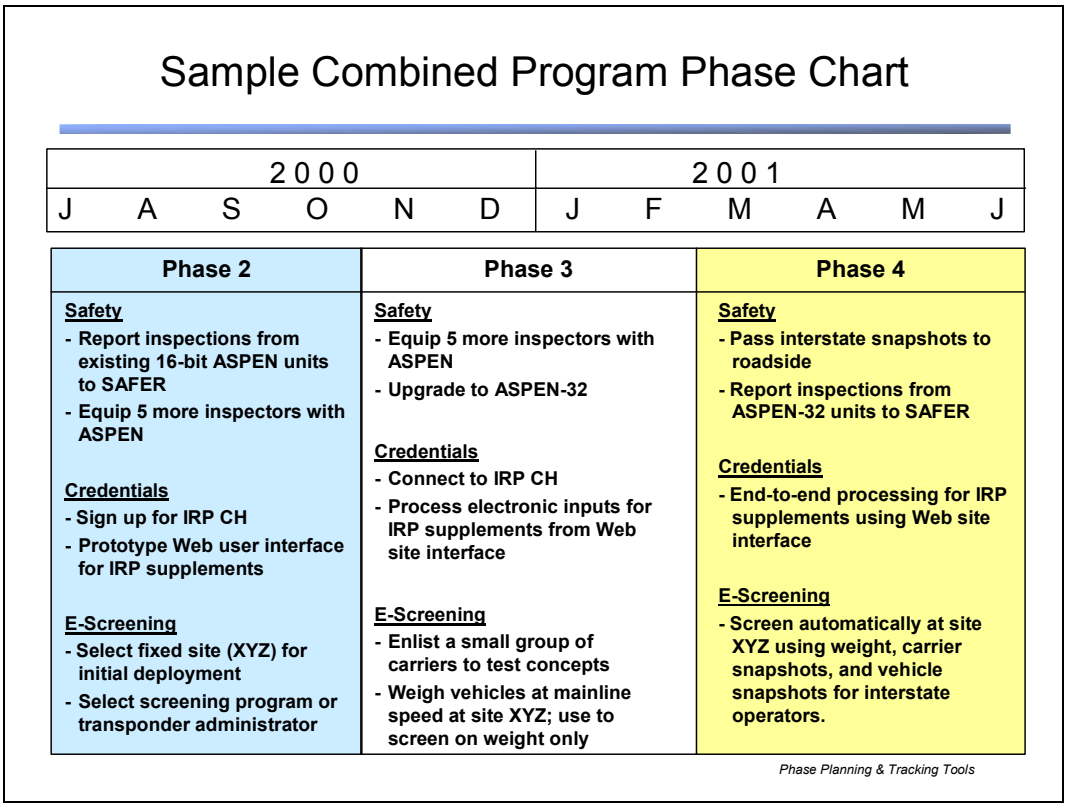


Figure B-2. Update the Phase Charts as Part of Detailed Scheduling

B.3.3. Detailed Schedule

The detailed schedule evolves from the target capabilities portrayed in the phase charts. The schedule should show the start and stop dates for the design, development, test, procurement, and integration activities related to accomplishing and demonstrating the phase objectives. Add external dependency tasks as needed.

There is no substitute for a detailed working-level schedule. How detailed? Down to about one person working for 2-4 weeks. At this point the team leader can assign real people to real tasks and get real work done, and at the same time can easily see if they are meeting the overall plan. Working-level schedules are "owned" by the development team leaders, which means they create and maintain them not merely as reporting vehicles but as tools that help get the job done.

The link to the phase charts is the vertical traceability downward of programmatic milestones (such as the end of a phase) and upward of strategic and visibility milestones.

B.3.4. Milestones

Milestones are expressed in a list using a planned date and a statement of the target accomplishment. As a phase proceeds, it's useful to indicate progress by showing the latest estimated completion date, and a measurement of how likely it is that the new date will be met. Don't eliminate the original planned date; then you can use the planned versus actual experience to do a better job of planning the next time.

Identify strategic and visibility milestones from each development team's detailed schedules. Strategic milestones are those that impact another development team or the end user, such as release of a new version of a software product. Visibility milestones are those that indicate tangible interim progress, such as release of a draft design document. Then consider how all the products will come together to be integrated, tested, and released for production use. Don't forget support tasks staffed by non-project team members (for example, operational staff conducting acceptance tests).

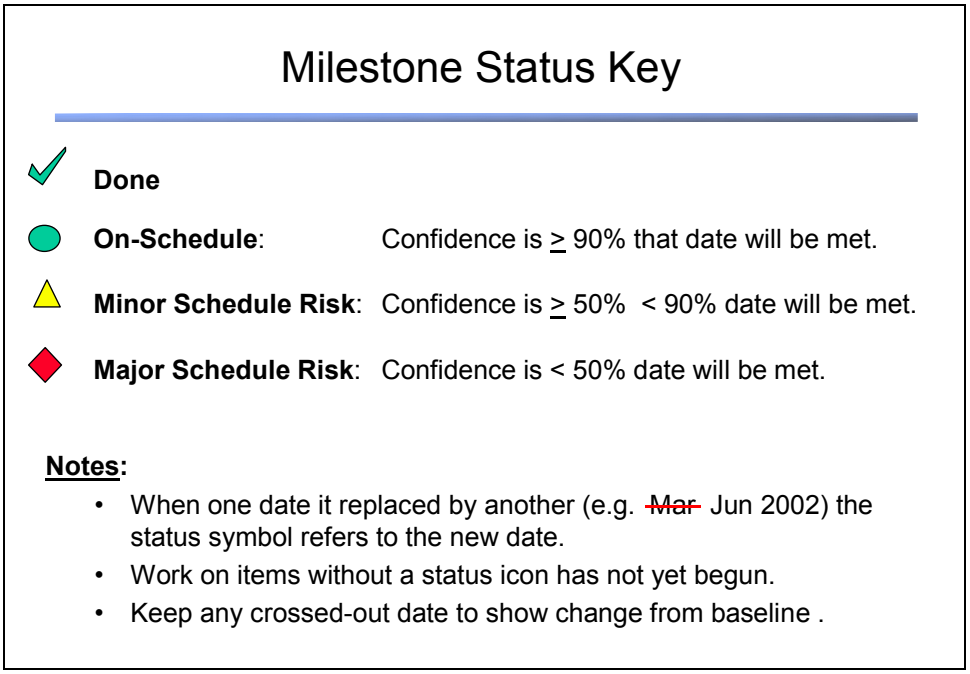


Figure B-3. Milestone Red-Yellow-Green Key

B.3.5. Budget

Budget plans are usually expressed as planned labor hours per unit time (often per month), and planned travel costs, material costs, subcontract costs, and other direct costs. The cost breakdown elements you plan and track will be dictated by your accounting system. You should also keep in mind what the funding sources expect in terms of accountability. The Phase Plan should indicate funds on hand and how they are expected to be spent over time.

B.3.6. Status Reports

A good test for what belongs in the Phase Plan is whether the charts and tables are in a format useful for presentation at monthly internal status meetings or quarterly external status meetings. An example would be a software version delivery schedule. As a result of being updated for these meetings, the Phase Plan itself is simultaneously kept up to date without additional effort.

In the words of one project manager, "the Phase Plan should be so useful that you actually feel like keeping it up to date."

Examples of status charts are given in Appendices C and D. They can be conveniently 3-hole punched and placed in the Project Leader's Notebook so you're prepared for an on-the-spot briefing. Note that large arrows are used to call attention to key points during an oral presentation.

B.4 Tab 4 – Issues, Action Items, Decisions

Continue to maintain the running list of issues, action items, and decisions. It might be useful to categorize the open items regarding which should be settled in the current phase, and which can be postponed. Figure B-3 shows how initially-vague concerns can eventually be settled as more information becomes available and people can work on them.

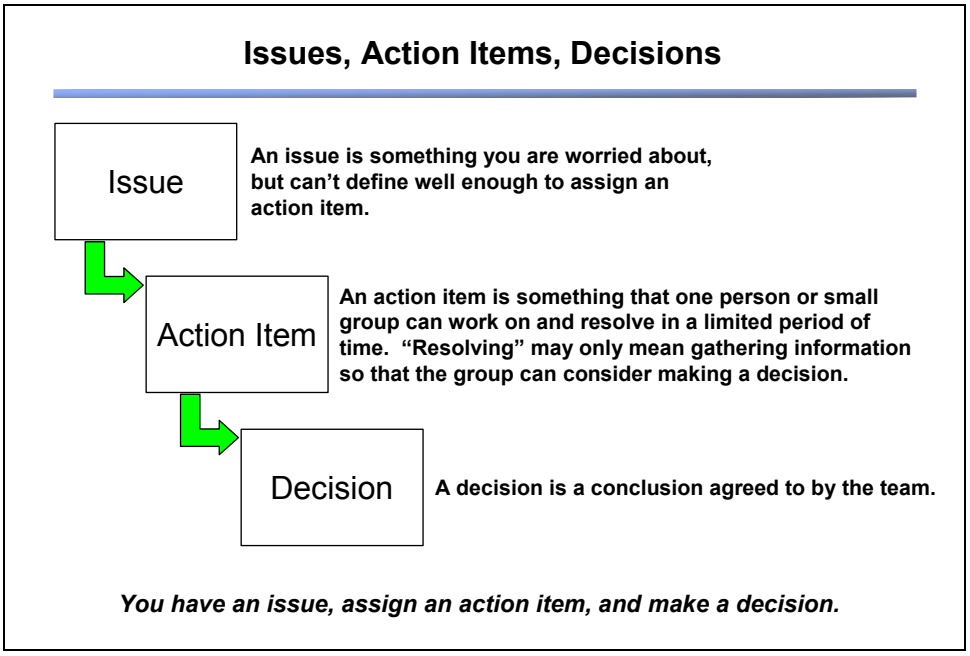


Figure B-4. Key Categories for Capturing Programmatic Information

B.5 Other Information

It may go without saying but we'll say it anyway: keep conveniently at hand an up-to-date copy of any planning and tracking elements that you truly find useful. Conversely, ignore what you don't find useful.

B.6 Do I Have Any Flexibility?!

The complexity and length of a Phase Plan will vary with the circumstances of the CVISN program in your state.

Overemphasis on documentation is not the answer, and can detract from developing a truly useful Phase Plan. You will have to work hard to find the right balance point for the team between "planning" and what they [unfortunately] perceive as the opposite of planning, "working".

APPENDIX C. SAMPLE MONTHLY PROJECT STATUS REPORT

Below is a sample Monthly Report. This is a illustrative only, not a real report. The target audience members are the leaders of other related CVISN projects within the organization. The oral presentation takes 10 minutes to deliver, and along with 13 other presentations is part of a monthly 2-1/2 hour meeting. For this peer-level audience the level of detail is finer than the example given in Appendix D. In the days immediately following the meeting, the organization's CVISN Program Manager extracts from these presentations and creates the monthly letter report that is a contractual deliverable.

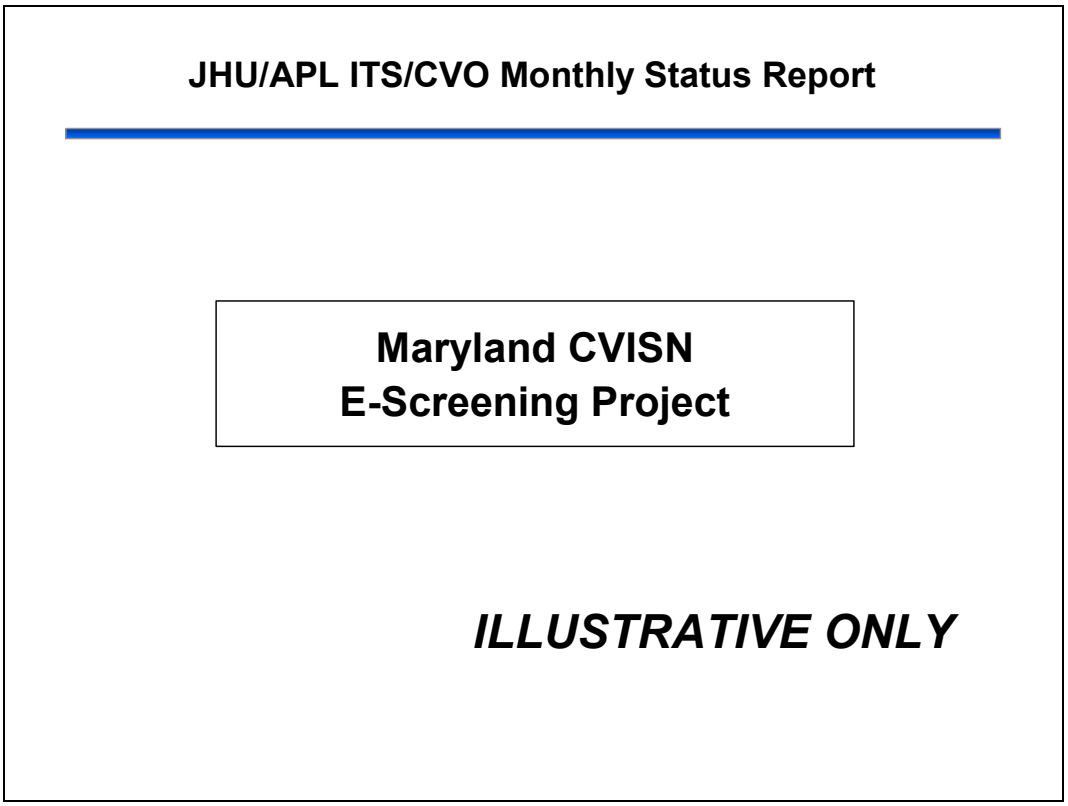


Figure C-1. Sample Project Monthly Report Format

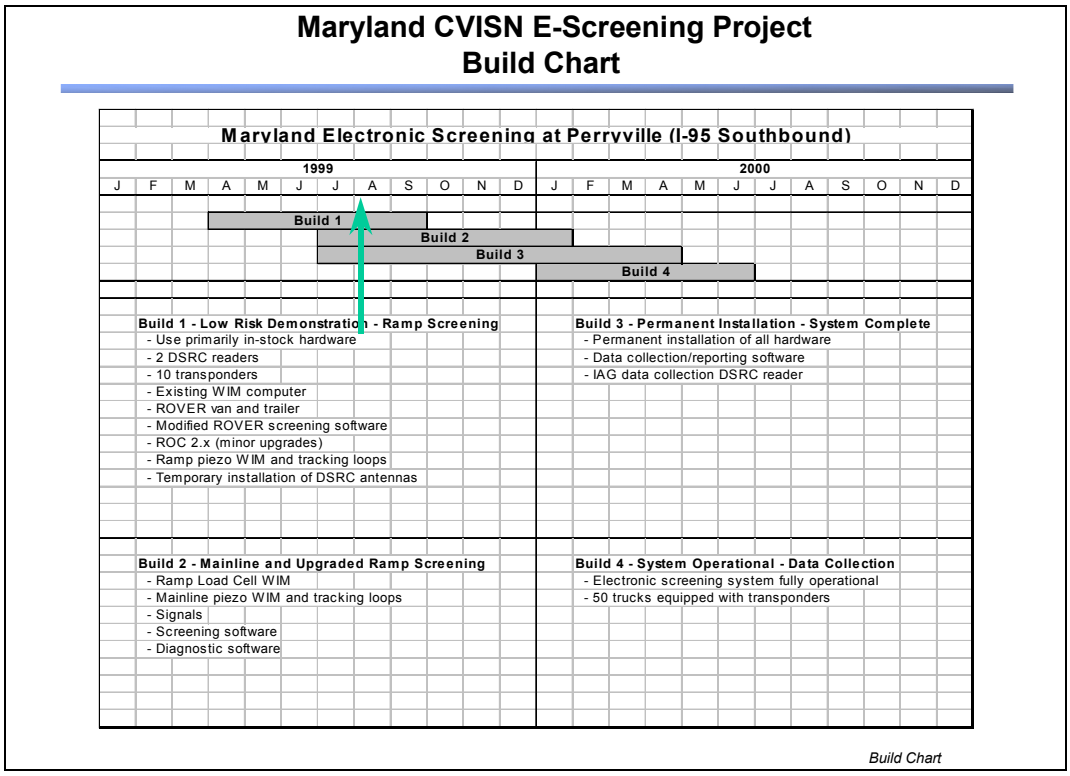


Figure C-2. Build Chart – Vertical Arrow Marks "Data Date"

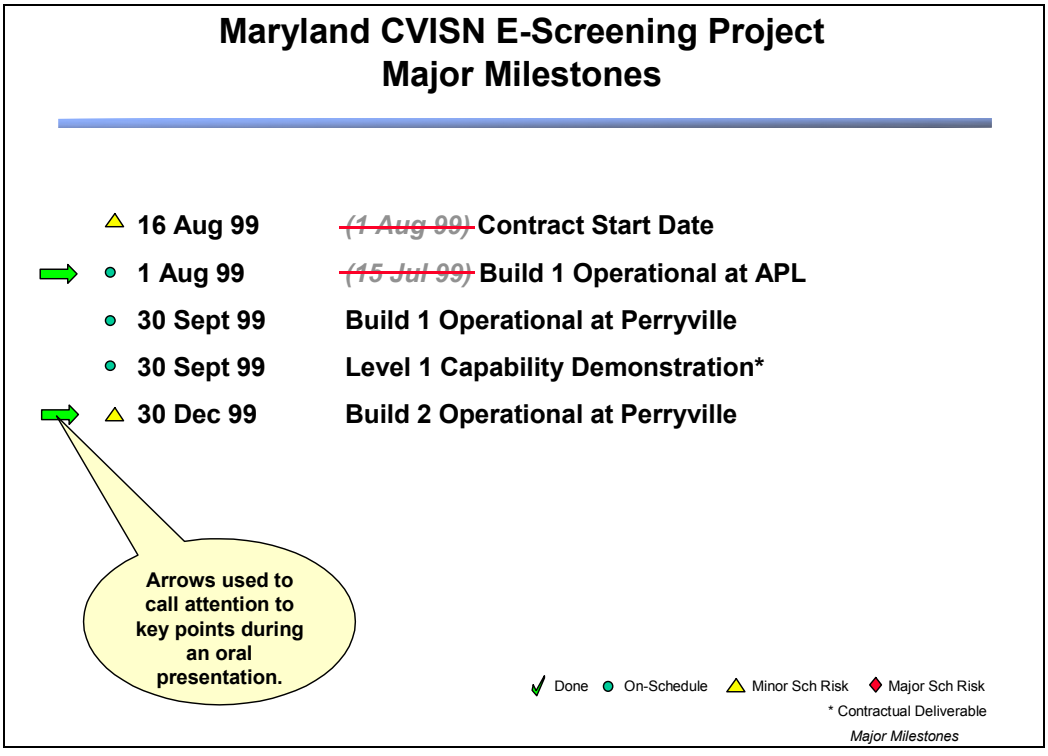


Figure C-3. Milestone Chart– Horizontal Arrows Flag Discussion Points

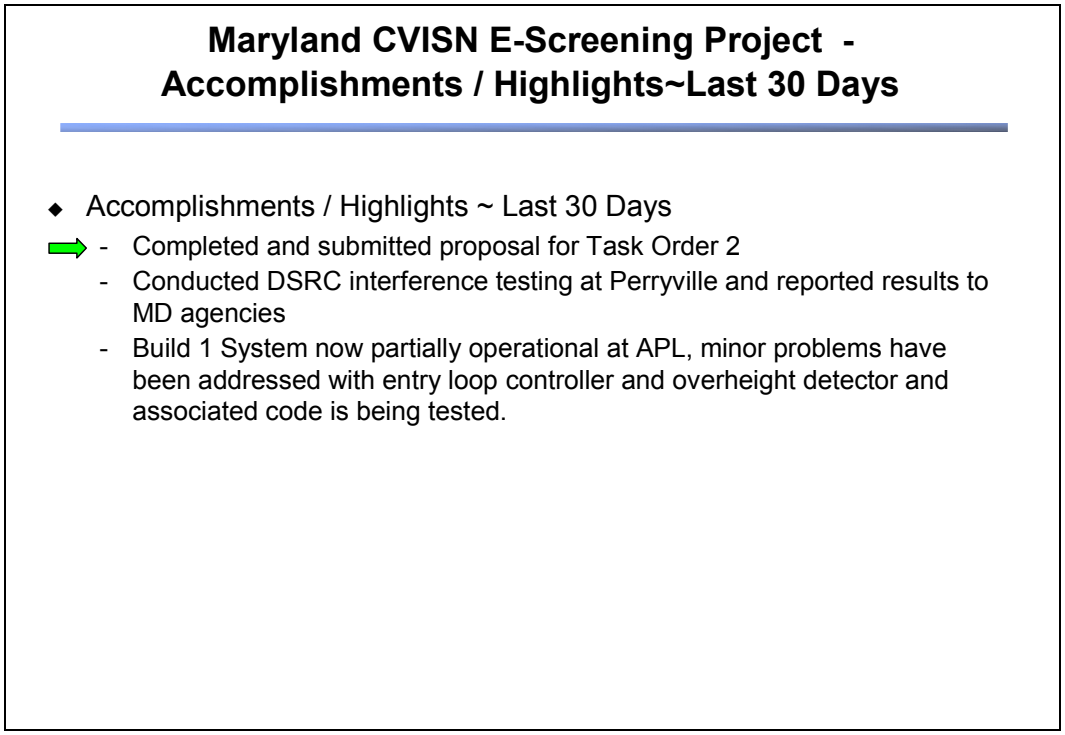


Figure C-4. Accomplishments – Horizontal Arrows Flag Discussion Points

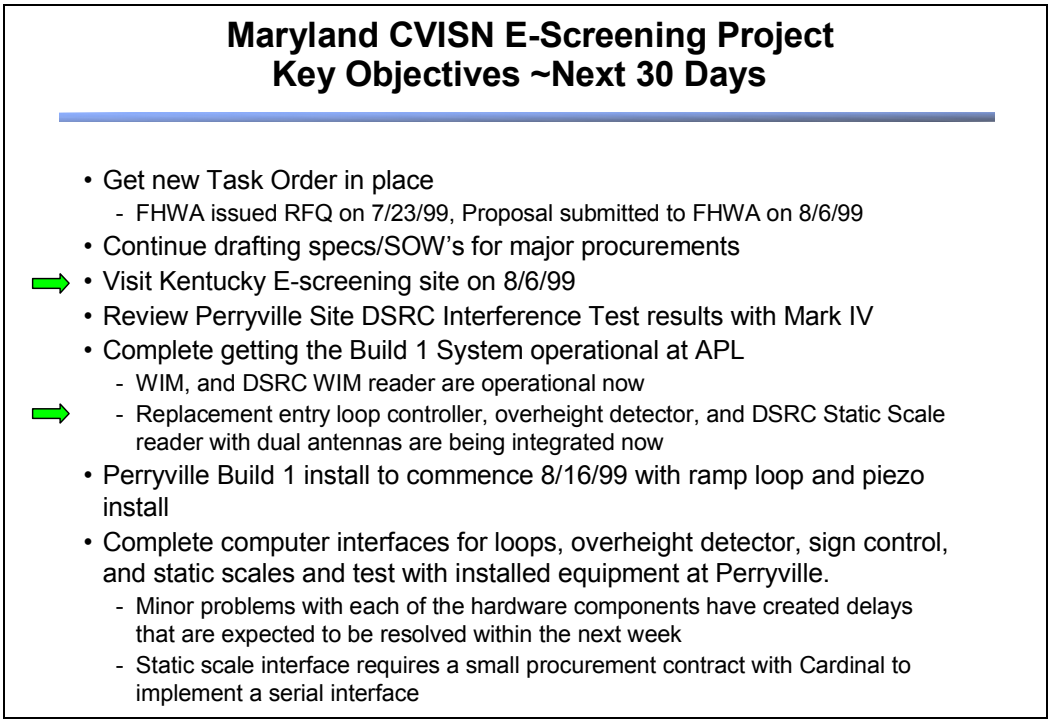


Figure C-5. Activities Next Month – Horizontal Arrows Flag Discussion Points

Maryland CVISN E-Screening Project
Current Issues

OPEN • Contract Start Date likely to be 16 August, project completion will not occur until 15 November 2000

- Procurements make up two thirds of the budget.
- A number of items (e.g., readers) have 16 week lead times; need to get orders in since we have to operate with borrowed assets until they arrive

OPEN • DSRC Interference with the MTAG system

- Continue to address concerns
- Testing of our transmitted field strengths has been conducted.
- MVA,MDOT,MdTA, SHA have directed (8/3/99) APL to liaison with Mark IV to attempt to resolve

Figure C-6. Issues – Horizontal Arrows Flag Discussion Points

Maryland CVISN E-Screening Project

Cost	Sched	Tech		*WBS Item*	Cost	Sched	Tech		*Notes*
G	Y	G	**1**	**Management**					
			1.6.1	Build 1 Procurement	Y	G	G	unplanned	Purchased replacement piezo WIMs, unplanned
			1.6.2	Build 2 Procurement	Y	Y	G	Aug-Nov-99	Awaiting new Task Order, expected about 16 August 99
			1.5.2	Level 1 Demonstration at Perryville	G	G	G	Sep-99	This will probably actually occur in October
G	G	G	**2**	**Systems Engineering/Studies/Demos**					
			2.1.1	Assess Compatibility with Existing Site Systems	G	G	G	Jun-99	Complete, Need follow with purchase contract to Cardinal to develop a static scale serial interface and to assisit us with interface to load cell WIM and ramp signs
			2.1.5	Assess DSRC Interference Issues	G	G	G	Jun-99	Complete
G	G	G	**3**	**Ramp Screening**					
			3.1.1	Reestablish Rover Capabilities	G	G	G	Jun-99	Complete, all subsystems operational
			3.1.4	Rover System Testing at APL	G	Y	G	~~Jul-99~~	(Aug 99)Some delays due to problems with loop controller and overheight detector; final testing at APL early August
G	G	G	**4**	**Mainline Screening**					
			4.1	Incorporate Additional Equipment Interfaces	G	G	G	~~Jul-99~~	(Oct 99) Mainline work is being rescheduled
			4.1.6	Update Screening Decisions	G	G	G	~~Jul-99~~	(Oct 99) Mainline work is being rescheduled
G	G	G	**5**	**Roadside Operations Computer**					
			5.1.1	ROC 2.0	G	G	G	May-99	Installed and operational in Rover
G	G	G	**6**	**Site Work**					
			6.1.1	Obtain Site Drawings	G	G	G	May-99	Complete
			6.1.2	Build 1 Site Planning	G	G	G	Jun-Jul 99	Prelim approval with MdTA
			6.2	Develop and Test Interfaces to Existing Equipment	G	G	G	Jul-99	Minor slip in schedule getting info from Cardinal
			6.3.1	Build 1 Installation at Perryville	G	G	G	Jul-Aug-99	
			6.3.2	Build 2 Installation at Perryville	G	Y	G	~~Sep-Nov 99~~	(Oct 99- Mar 00) Reflects late project start

Figure C-7. WBS Annotated for Technical, Schedule, and Cost Status

APPENDIX D. SAMPLE QUARTERLY PROGRAM STATUS REPORT

Below is a sample Quarterly Report prepared by the Program Manager one of the CVISN Pilot States. It is at the level of an executive briefing, covering all the projects that make up the CVISN program in the state. The target audience members were FMCSA and the CVISN state Program Managers. It is can be delivered orally or distributed via e-mail.

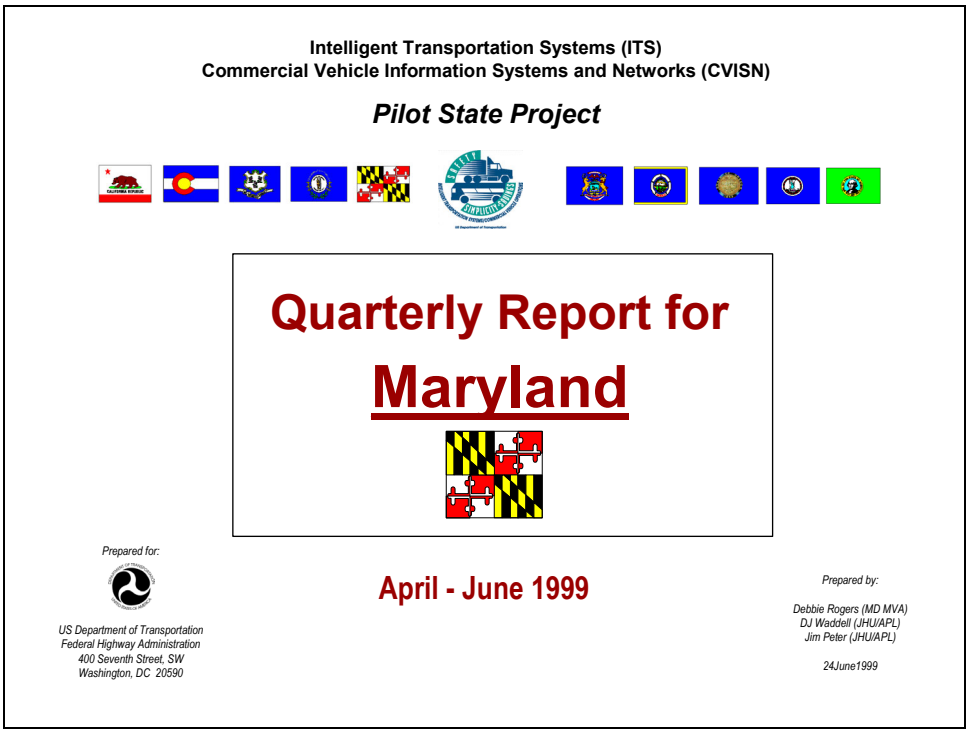


Figure D-1. Sample State Quarterly Report Format

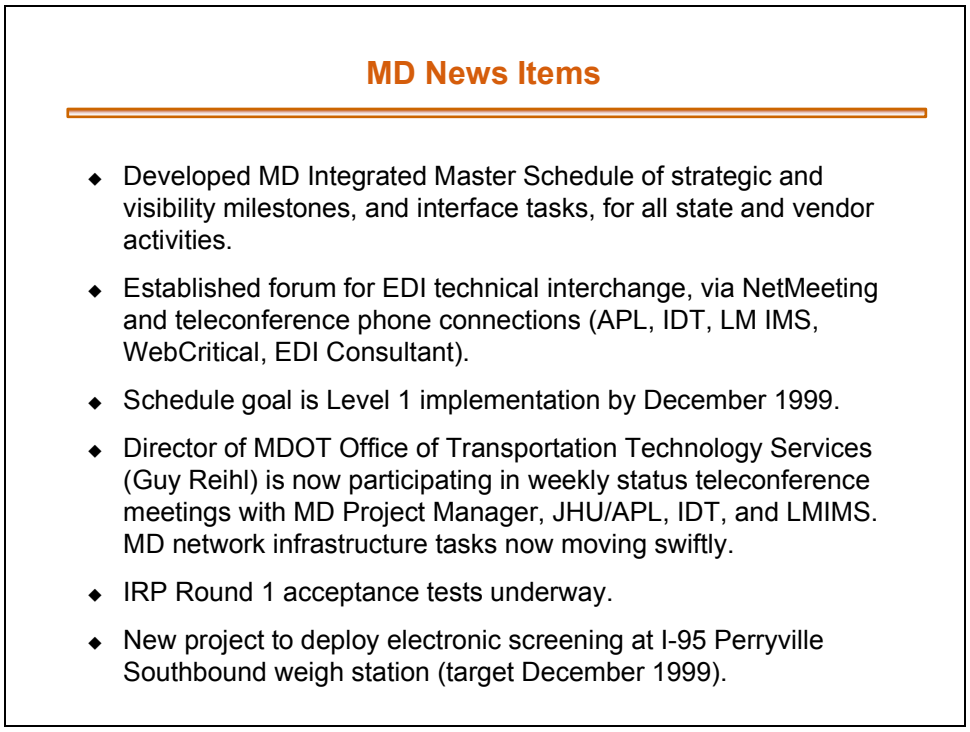


Figure D-2. Presentation Leads Off with News Bulletins

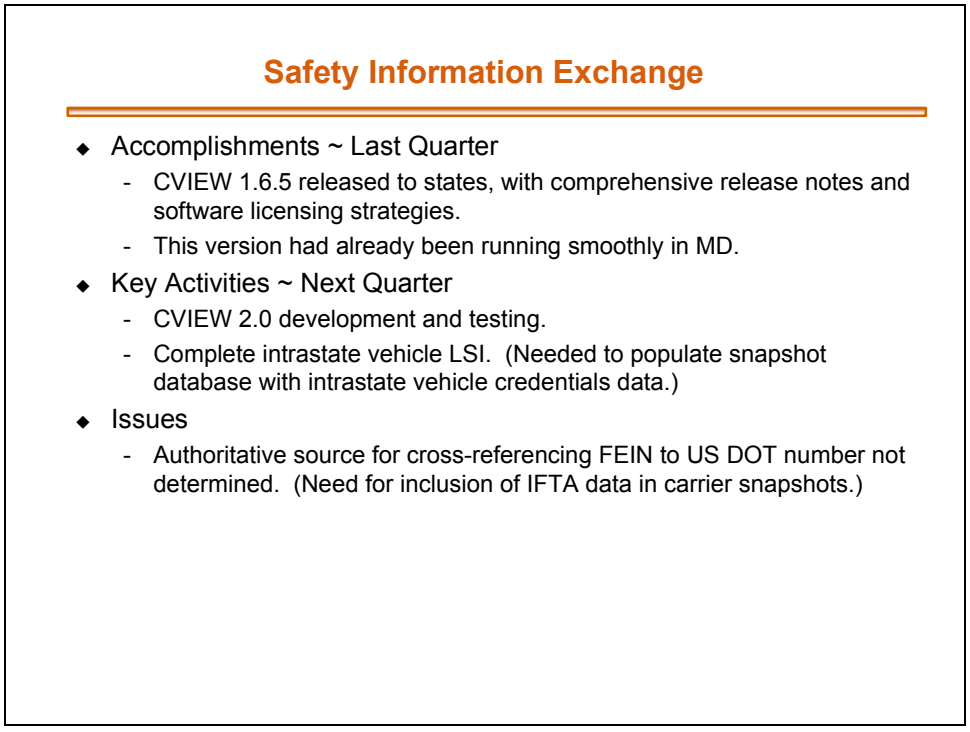


Figure D-3. Accomplishments, Plans, and Issues for Safety Information Exchange

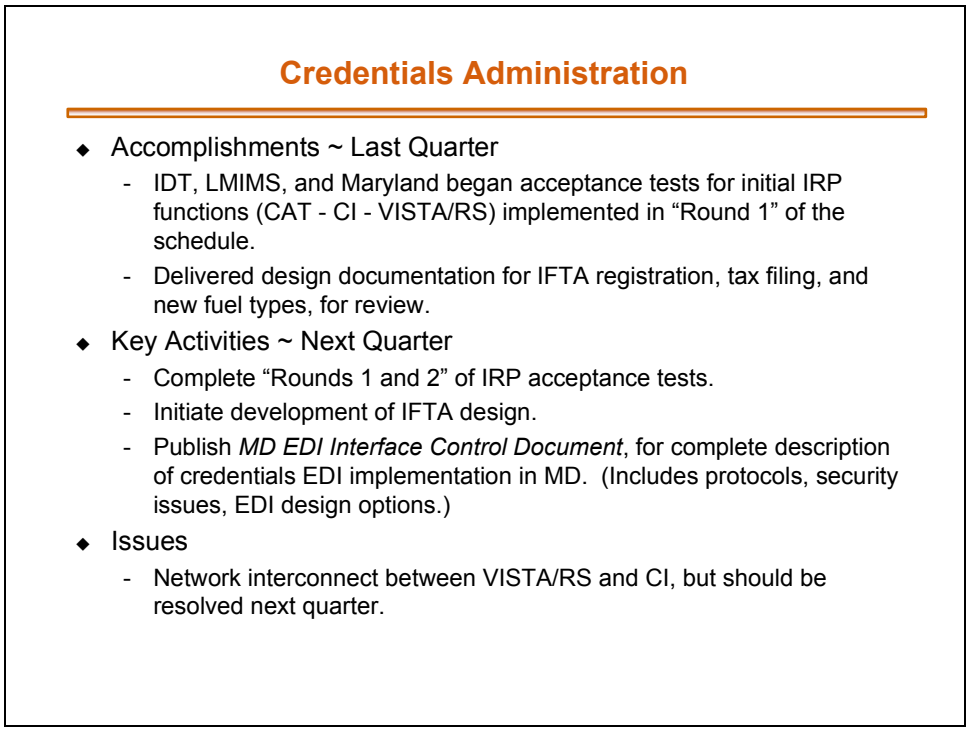


Figure D-4. Accomplishments, Plans, and Issues for Credentials Administration

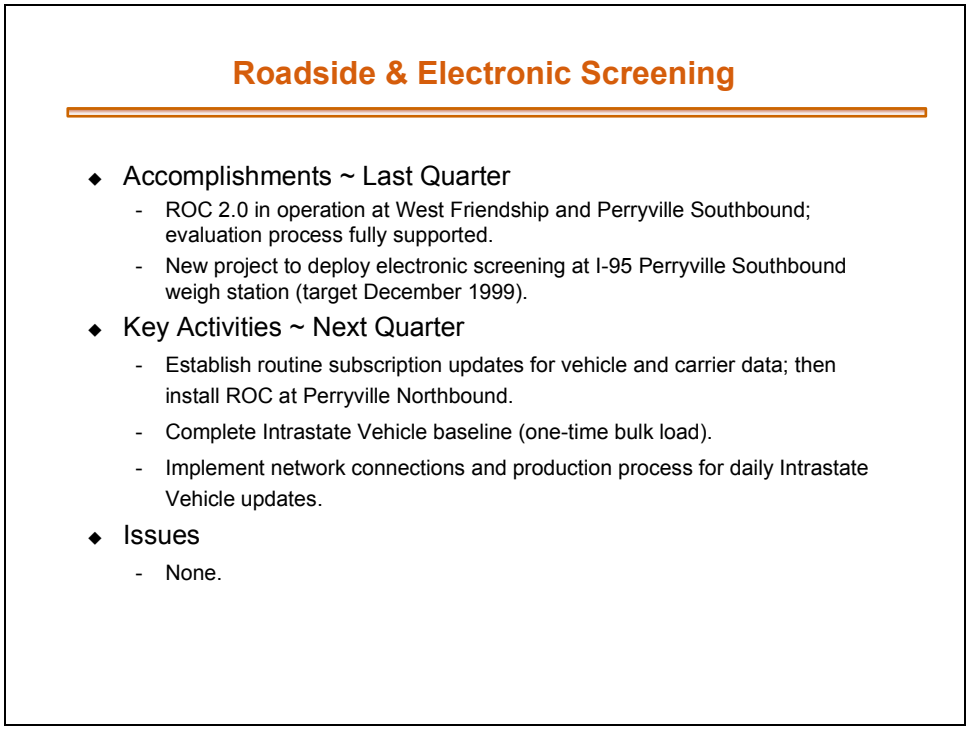


Figure D-5. Accomplishments, Plans, and Issues for Roadside & Electronic Screening

Project Management & System Engineering

- Accomplishments ~ Last Quarter
  - Refined, updated, and promulgated Critical Path Network and Gantt chart for the MD CVISN Master Integrated Schedule.
  - Gained full participation on project planning and staff support from MDOT Office of Transportation Technology Services, for network infrastructure tasks .
  - MVA issued Purchase Order for MD CVISN Server and Network upgrades.
- Key Activities ~ Next Quarter
  - Define scope of work for NLETS interface to ROC or CVIEW.
  - APL, MVA, MDOT, and CSC plan for configuring development and test environment, and "warm spare" back-up for production environment.
- Issues
  - None.

Figure D-6. Accomplishments, Plans, and Issues for Project Management & System Engineering

Maryland CVISN Phases

1998												1999												2000											
J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D	J	F	M	A	M	J	J	A	S	O	N	D
						Phase 5						Phase 6						Phase 7						Phase 8											

Phase 5	Phase 6	Phase 8
• ROC 1.1 at West Friendship with CVIEW interface • CVIEW 1.6 operational: interstate carrier and vehicle safety data • CAT to carriers for IRP submission • CAT to IFTA Office for testing • CAT & CI initial operating capability • CI User Interface with access control for IRP vs IFTA staff	• Install ROC at Perryville • Evaluate ROC at West Friendship & Perryville • IRP Flat File - CVIEW Update • CI-MVA Mainframe Interface for Flags & Conditions • Initial IRP processing end-to-end • Intrastate vehicle - CVIEW update • CVIEW 2.0 available • CAT to IFTA Jessup field office • CI-RPC IFTA Tax Filing submission	• CVIEW 4.0 available • Perryville E-screening – Build 3 permanent poles and signals • Intrastate vehicle credentialing complete • Perryville E-screening – Build 4 handover operational system • Analyze electronic screening systems & generate requirements for West Friendship
	Phase 7 • Perryville E-screening - Build 1 ramp temporary • CVIEW 3.0 available • IRP processing complete • Perryville E-screening - Build 2 mainline/permanent ramp • Intrastate vehicle credentialing initial operating capability	

Figure D-7. Maryland Phase Chart

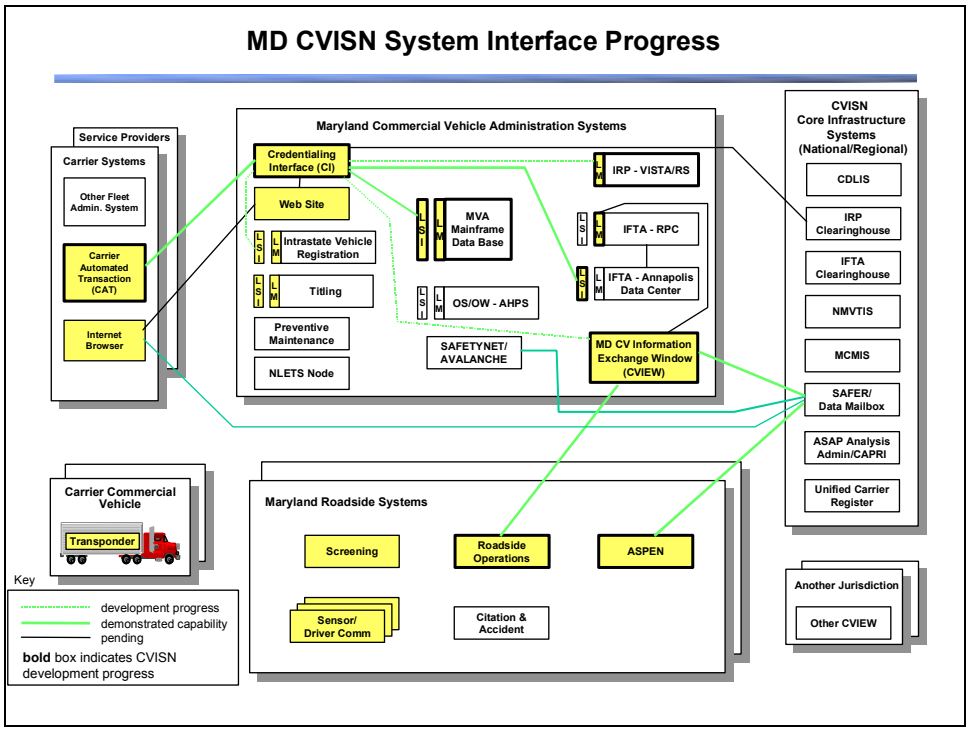


Figure D-8. System Interface Diagram with Progress Noted

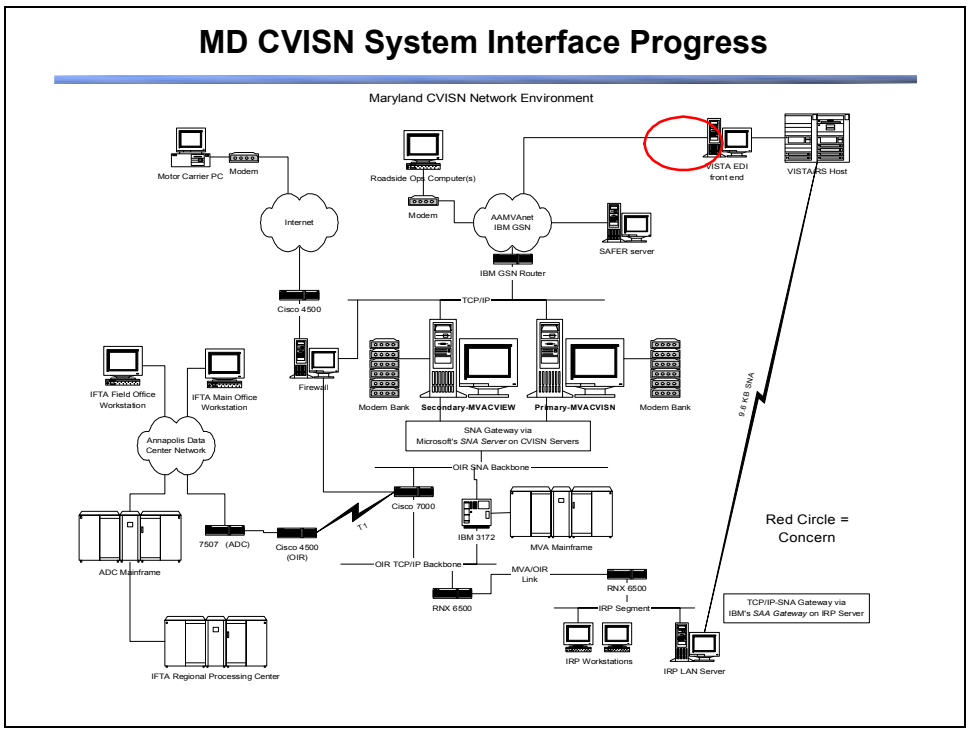


Figure D-9. A Different View of the System, with Issue Noted

MD CVISN Work Breakdown Structure (1 of 3)

Cost	Sched	Tech		WBS Item	Cost	Sched	Tech	Dates	Notes
			1	Project Management					
			1.1	Project Plan (revise)		Y			
			1.2	Phase Plans					
			1.3	Integrated Master Schedule					
			1.4	Review Subsystem Detailed Development Schedules					
			1.5	Project Progress Facilitation (Action Item lists, etc)					
			1.6	Liason & Coordinaton with Agencies (FHWA, MDOT, IACG, MVA, SHA, MdTA, MDSP)					
			1.7	Liason & Coordination with Contractors (IDT, LMIMS)					
			1.8	Liason & Coordination with Related Projects (CVIEW, SAVER, ASPEN)					
			1.9	Status Reporting (APL monthly, MD/VA quarterly, FHWA quarterly)					
			1.10	Subcontract Management (IDT, procurements)					
			1.11	Presentations at Conferences & Seminars					
			2	System Engineering					
			2.1	System Design Description (revise)		Y			
			2.2	Requirements Analysis					
			2.3	Support Interoperability Test Development					Includes data sets (example: ficticious carriers), and scenarios.
			2.4	Support Acceptance Test Development					Includes data sets (example: ficticious carriers), and scenarios.
			2.5	MD Credentialing EDI Interface Control Document					
			2.6	Networking Issues (security, protocols, connectivity)					
	Y		3	Credentials					
			3.1	VISTA/RS Interface		Y			
			3.2	IRP LSI					
			3.3	Intrastate Reg & Titling LSI (ultimately bi-directional)					
			3.4	IFTA RPC Interface					
			3.5	IFTA Reg LSI					
			3.6	InterCAT					
			3.7	CI		Y			
			3.8	CyberCAT (including RECON server interface)					
			3.9	MVA Mainframe DB Retrieval / Update					CI needs to update mainframe, like IRP flat file does now.
			3.10	IRP Deployment					
			3.10.1	CAT for All IRP staff					Replaces 3270-emulation mode.
			3.10.2	CI User Interface for IRPAdministrators					Only Sharon Hopkins has this now.
			3.11	IFTA Deployment					
			3.11.1	CAT on One System for MFTU Staff					Replaces 3270-emulation mode.
			3.11.2	CAT on One System for Jessup Field Office Staff					Replaces 3270-emulation mode.
			3.11.3	CI User Interface for IFTA Administrators					IFTA office has this.
			3.12	Intrastate Registration Deployment					
			3.13	Intrastate Titling Deployment					

Note: See MD Master Integration Schedule for all dates.

Figure D-10. MD WBS Annotated for Technical, Schedule, and Cost Status

MD CVISN Work Breakdown Structure (2 of 3)

Cost	Sched	Tech		WBS Item	Cost	Sched	Tech	Dates	Notes
			4	Electronic Screening					
			4.1	ROC					
			4.2	Perryville Site Deployment					E-Screening being developed.
			4.3	West Friendship Site Specifications					Ron Char will continue to develop the spec.
			5	Safety Information Exchange					
			5.1	CVIEW					
			5.2	Intrastate Vehicle LSI					
			5.3	IRP LSI				Complete	VISTA/RS flat file created manually by MVA IRP staff, then sent to MVA mainframe. Also sent manually via FTP to CVIEW host. Needs to be automated. Desire flat file created automatically but at least simplify so it only needs to be sent manually once.
			6	Electronic Funds Transfer					Capability
	Y		7	Network Infrastructure					
			7.1	General Configuration, Growth Stategy, Migration Path					
			7.2	MDOT SNA Network (Mainframe)					Ops & maintenance.
			7.3	MDOT TCP/IP Network					Ops & maintenance.
			7.3.1	Dual-Homing (Internet & AAMVAnet)		Y			Stalled, possibly due to new firewall currently being installed.
			7.3.2	Access Control					
			7.3.3	Virtual Private Networking		Y			Could be PPTP (Microsoft proprietary), or IPSec (open).
			7.4	AAMVAnet					
			7.4.1	Long-Term State Contract					
			7.4.2	Local Dial-Up Connectivity from ASPEN Sites		Y			
			7.4.3	Op & Maint of Network Node					Router for AAMVAnet frame relay network, located at MVA. Administrative tasks for the IBM Global Svcs account.

Note: See MD Master Integration Schedule for all dates.

Figure D-11. MD WBS Annotated for Technical, Schedule, and Cost Status

MD CVISN Work Breakdown Structure (3 of 3)

Cost	Sched	Tech		WBS Item	Cost	Sched	Tech	Dates	Notes
			8	Acceptance Tests					
			8.1	MD Staff Involvement for Review, Evaluation, Acceptance					
			8.2	CAT-CI-VISTA/RS Round 1					One of these must include CI-mainframe flags & conditions. Also CI-CVIEW.
			8.3	CAT-CI-VISTA/RS Round 2					
			8.4	CAT-CI-VISTA/RS Round 3					
			8.5	CAT-CI-VISTA/RS Round 4					
			8.6	[obsolete]					
			8.7	Perryville Build 1					
			8.8	Perryville Build 2					
			8.9	Perryville Build 3					
			8.10	Perryville Build 4					
			9	Interoperability Tests					Assume Mary Stuart is Test Conductor. Not to be confused with Interop Test project at APL. We use the plans/procedures produced by that project to conduct these tests.
			9.1	Conduct Part 1					
			9.2	Report Part 1					
			9.3	Conduct Part 2					
			9.4	Report Part 2					
			9.5	Conduct Part 2					
			9.6	Report Part 2					

Note: See MD Master Integration Schedule for all dates.

Figure D-12. MD WBS Annotated for Technical, Schedule, and Cost Status

MD COACH Part 1

Commit Level (F/P/N)	Item #	Compatibility Criteria	Req Level (L1/E/C)	Op Test Date	IOC Date	FOC Date	Comments
F	**5.1.1**	Adopt standard identifiers for carriers, vehicles, drivers, and transponders to support information exchange	L1	*ignore*	*ignore*	*ignore*	Need USDOT source for intrastate carriers
F	**5.1.2**	Use open standards for exchange of information with other jurisdictions and with the public.	L1	*ignore*	*ignore*	*ignore*	
F	**5.1.3**	Ensure that all information transfers, fee payments, and money transfers are authorized and secure.	L1	*ignore*	*ignore*	*ignore*	
F	**5.1.4**	Exchange safety and credentials data electronically within the state to support credentialing, safety, and other roadside functions. Where useful, exchange snapshots.	L1	*ignore*	Feb-99	Dec-99	Snapshots available at West Friendship weigh station.
F	**5.1.5**	Demonstrate technical interoperability by performing Interoperability Tests.	L1	*ignore*	Jul-99	Dec-99	
F	**5.1.6**	Support electronic payments.	E	*ignore*	*ignore*	*ignore*	

Figure D-13. MD COACH Part 1 Tables

MD COACH Part 1

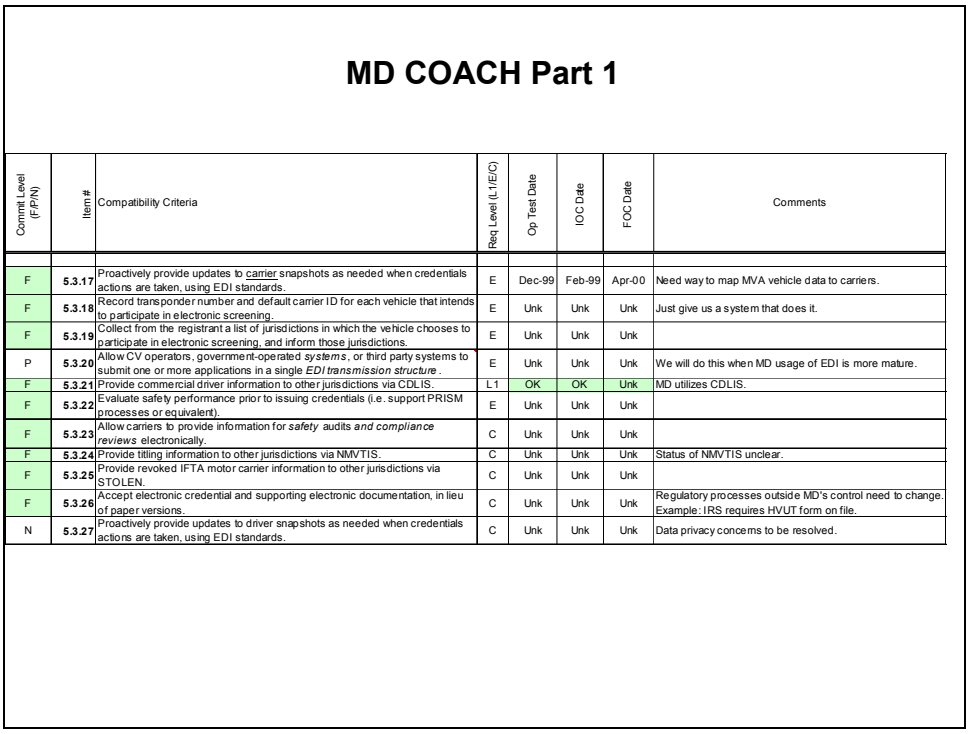
Commit Level (F/P/N)	Item #	Compatibility Criteria	Req Level (L1/E/C)	Op Test Date	IOC Date	FOC Date	Comments
F	**5.2.1**	Use ASPEN (or equivalent) at all major inspection sites	L1	~Oct-98	Apr-99	Unk	JJP will confirm dates with P.North
F	**5.2.2**	SAFETYNET 2000 submits interstate and intrastate inspection reports to SAFER.	L1	Unk	Unk	Unk	MD waiting to receive. JJP will confirm with P.North.
P	**5.2.3**	Use CAPRI (or equivalent) for compliance reviews. *(Indicate what your state uses.)*	L1	Unk	Unk	Unk	MD has its own preventive maint program.
N	**5.2.4**	Collect, store, analyze, and distribute citation data electronically.	E	N/A	N/A	N/A	
F	**5.2.5**	Collect, store, analyze, and distribute crash data electronically.	E	Unk	Unk	Unk	Will support when available.
N	**5.2.6**	Compute carrier safety risk rating for intrastate carriers based on safety data collected.	E	N/A	N/A	N/A	No intrastate operating licensing req'd in MD.
F	**5.2.7**	Identify high-risk drivers based in the state through regular performance evaluation of various factors such as license status, points, and inspections.	C	Unk	Unk	Unk	MD participates in CDLIS.

Figure D-14. MD COACH Part 1 Tables

MD COACH Part 1

Commit Level (F/P/N)	Item #	Compatibility Criteria	Req Level (L1/E/C)	Op Test Date	IOC Date	FOC Date	Comments
F	**5.3.1**	Support electronic credentialing (electronic submission of applications, evaluation, processing, and application response) for IRP using EDI standards.	L1	Unk	Jun-98	Dec-99	2 large carriers (Rollins & Penske) using routinely, 4 others occasionally.
F	**5.3.2**	Proactively provide updates to vehicle snapshots as needed when IRP credentials actions are taken, using EDI standards.	L1	Unk	Nov-98	Jan-99	LSI flat file to CVIEW; CVIEW to SAFER is EDI. (Does LSI gen 285?)
F	**5.3.3**	Proactively provide updates to carrier snapshots as needed when IRP credentials actions are taken, using EDI standards.	L1	Unk	Unk	Unk	USDOT number not reliably captured; cross ref vehicle to carrier not established.
F	**5.3.4**	Provide IRP Clearinghouse with IRP credential application information (recaps).	L1	Unk	Unk	Dec-98	
F	**5.3.5**	*Use data provided by the IRP Clearinghouse to review fees billed and/or collected by a jurisdiction and the portion due other jurisdictions (transmittals).*	L1	Apr-98	Apr-98	Apr-98	MD in first wave of states to use Clearinghouse.
F	**5.3.6**	Support electronic state-to-state fee payments via IRP Clearinghouse.	L1	Nov98?	Nov98?	Nov98?	JJP check with S.Hopkins
F	**5.3.7**	Support electronic credentialing (electronic submission of applications, evaluation, processing, and application response) for IFTA registration using EDI standards.	L1	Nov-99	Dec-99	Apr-00	
F	**5.3.8**	Proactively provide updates to carrier snapshots as needed when IFTA credentials actions are taken or tax payments are made, using EDI standards.	L1	Oct-99	Dec-99	Apr-00	
F	**5.3.9**	Provide IFTA Clearinghouse with IFTA credential application information using EDI standards.	L1	Unk	Unk	Unk	IFTA RPC is MD's agent now. Clearinghouse may not accept EDI.
F	**5.3.10**	Support electronic tax filing for IFTA quarterly fuel tax returns using EDI standards.	L1	Dec-99	Feb-00	Apr-00	
F	**5.3.11**	Provide information on taxes collected by own jurisdiction and the portion due other jurisdictions (transmittals) to the IFTA Clearinghouse using EDI standards.	L1	Unk	Unk	Unk	IFTA RPC is MD's agent now. RPC reports to Clearinghouse in legacy format.
F	**5.3.12**	Download for automated *(preferably)* review the demographic *(eg address)* information from the IFTA Clearinghouse using EDI standards. *(Public information.)*	L1	Unk	Unk	Unk	Is IFTA Clearinghouse inviting new states yet? What does "automated review" mean?
F	**5.3.13**	Download for automated *(preferably)* review the transmittal information from the IFTA Clearinghouse using EDI standards. *(Privileged information.)*	L1	Unk	Unk	Unk	
F	**5.3.14**	Retrieve IFTA tax rate information electronically from IFTA, Inc.	L1	Dec-99	Feb-99	Apr-00	
F	**5.3.15**	Support electronic credentialing (electronic submission of applications, evaluation, processing, and application response) for other credentials using EDI standards.	E	Unk	Unk	Unk	MD does not require registration for intrastate carriers, except for flammables and passenger carriers.
F	**5.3.16**	Proactively provide updates to vehicle snapshots as needed when credentials actions are taken, using EDI standards.	E	Dec-98	Feb-99	Mar-99	Already doing daily snapshot updates of IRP vehicle data.

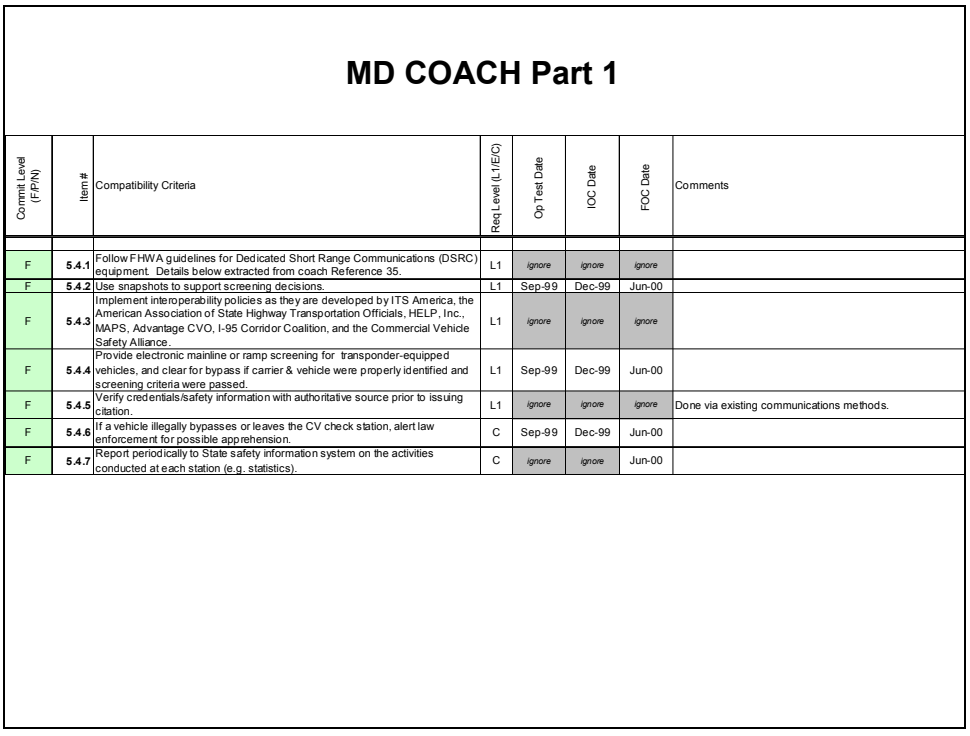
Figure D-15. MD COACH Part 1 Tables



MD COACH Part 1

Commit Level (F/P/N)	Item #	Compatibility Criteria	Req Level (L1/E/C)	Op Test Date	IOC Date	FOC Date	Comments
F	**5.3.17**	Proactively provide updates to carrier snapshots as needed when credentials actions are taken, using EDI standards.	E	Dec-99	Feb-99	Apr-00	Need way to map MVA vehicle data to carriers.
F	**5.3.18**	Record transponder number and default carrier ID for each vehicle that intends to participate in electronic screening.	E	Unk	Unk	Unk	Just give us a system that does it.
F	**5.3.19**	Collect from the registrant a list of jurisdictions in which the vehicle chooses to participate in electronic screening, and inform those jurisdictions.	E	Unk	Unk	Unk	
P	**5.3.20**	Allow CV operators, government-operated *systems*, or third party systems to submit one or more applications in a single *EDI transmission structure*.	E	Unk	Unk	Unk	We will do this when MD usage of EDI is more mature.
F	**5.3.21**	Provide commercial driver information to other jurisdictions via CDLIS.	L1	OK	OK	Unk	MD utilizes CDLIS.
F	**5.3.22**	Evaluate safety performance prior to issuing credentials (i.e. support PRISM processes or equivalent).	E	Unk	Unk	Unk	
F	**5.3.23**	Allow carriers to provide information for *safety* audits *and compliance reviews* electronically.	C	Unk	Unk	Unk	
F	**5.3.24**	Provide titling information to other jurisdictions via NMVTIS.	C	Unk	Unk	Unk	Status of NMVTIS unclear.
F	**5.3.25**	Provide revoked IFTA motor carrier information to other jurisdictions via STOLEN.	C	Unk	Unk	Unk	
F	**5.3.26**	Accept electronic credential and supporting electronic documentation, in lieu of paper versions.	C	Unk	Unk	Unk	Regulatory processes outside MD's control need to change. Example: IRS requires HVUT form on file.
N	**5.3.27**	Proactively provide updates to driver snapshots as needed when credentials actions are taken, using EDI standards.	C	Unk	Unk	Unk	Data privacy concerns to be resolved.

Figure D-16. MD COACH Part 1 Tables



MD COACH Part 1

Commit Level (F/P/N)	Item #	Compatibility Criteria	Req Level (L1/E/C)	Op Test Date	IOC Date	FOC Date	Comments
F	**5.4.1**	Follow FHWA guidelines for Dedicated Short Range Communications (DSRC) equipment. Details below extracted from coach Reference 35.	L1	*ignore*	*ignore*	*ignore*	
F	**5.4.2**	Use snapshots to support screening decisions.	L1	Sep-99	Dec-99	Jun-00	
F	**5.4.3**	Implement interoperability policies as they are developed by ITS America, the American Association of State Highway Transportation Officials, HELP, Inc., MAPS, Advantage CVO, I-95 Corridor Coalition, and the Commercial Vehicle Safety Alliance.	L1	*ignore*	*ignore*	*ignore*	
F	**5.4.4**	Provide electronic mainline or ramp screening for transponder-equipped vehicles, and clear for bypass if carrier & vehicle were properly identified and screening criteria were passed.	L1	Sep-99	Dec-99	Jun-00	
F	**5.4.5**	Verify credentials/safety information with authoritative source prior to issuing citation.	L1	*ignore*	*ignore*	*ignore*	Done via existing communications methods.
F	**5.4.6**	If a vehicle illegally bypasses or leaves the CV check station, alert law enforcement for possible apprehension.	C	Sep-99	Dec-99	Jun-00	
F	**5.4.7**	Report periodically to State safety information system on the activities conducted at each station (e.g. statistics).	C	*ignore*	*ignore*	Jun-00	

Figure D-17. MD COACH Part 1 Tables

This Page Intentionally Blank

APPENDIX E. HOW DO I RUN A PHASE PLANNING TEAM SESSION?

This appendix describes how you might conduct a phase planning session (or "mini-workshop") with your project team (working level) or program team (managers). Include all the programmatic and technical leaders for the project and phase.

Schedule it towards the end of the current phase. You'll need to have a clear idea of what objectives you will have accomplished by the end of the current phase. On the other hand you'll want to finish the planning session for the next phase before that phase starts so that you can focus on the technical work there as soon as the current phase is completed.

E.1 Advance Preparation Activities

Before you conduct a phase planning session with your team there are preparations to be made. Start far enough in advance to ensure that you have a successful meeting. The session location and schedule need to facilitate concentration on planning activities and minimize interruptions. The major preparation activities are outlined below.

Identify Facilitator

- The facilitator can be from the CVISN Team or can be an outside resource. A facilitator's role is to run the meeting so that the Project Leader can feel free to participate in the discussions about the project rather than be distracted with the process of the meeting itself.

Arrange Session Logistics

- Attendees – be sure to invite the leaders of each major effort.
- Invitations – written invitations are better than verbal ones where details can be forgotten; use e-mail or your organization's electronic calendar.
- Meeting Space:
  - A room to seat the entire group.
  - Tables and chairs or other arrangement that allows the group to optionally work in breakout subgroups.
  - Room equipped with computer and projector, screen, easels with paper, and markers.
  - Objectives, WBS, work assignments and phase schedule posted on walls and/or copies for all attendees.
  - Equipment and supplies for small group breakouts.
- Agenda – an example is provided later in this appendix.
- Schedule – no longer than 2 days; often one day or less.
- Break and meal supplies.

Notify Participants

- Advise participants of proposed schedule and agenda.
- Determine whether all participants will be needed in all of the sessions or whether there will be specified breakout sessions for subgroups of participants.
- Obtain confirmation of participation.
- Include technical resources outside the state CVISN team such as contractors, when appropriate. (Such resources could participate in the whole meeting or only in portions where they contribute technical and schedule information.)

Distribute Pre-Work

- Develop and distribute whatever pre-work materials and instructions that will help kick-start the session itself.
- Identify any up-front analysis that must be completed before the session begins.

E.2 Phase Planning Session Activities

The phase planning session consists of three main stages, as shown in Figure E-1.

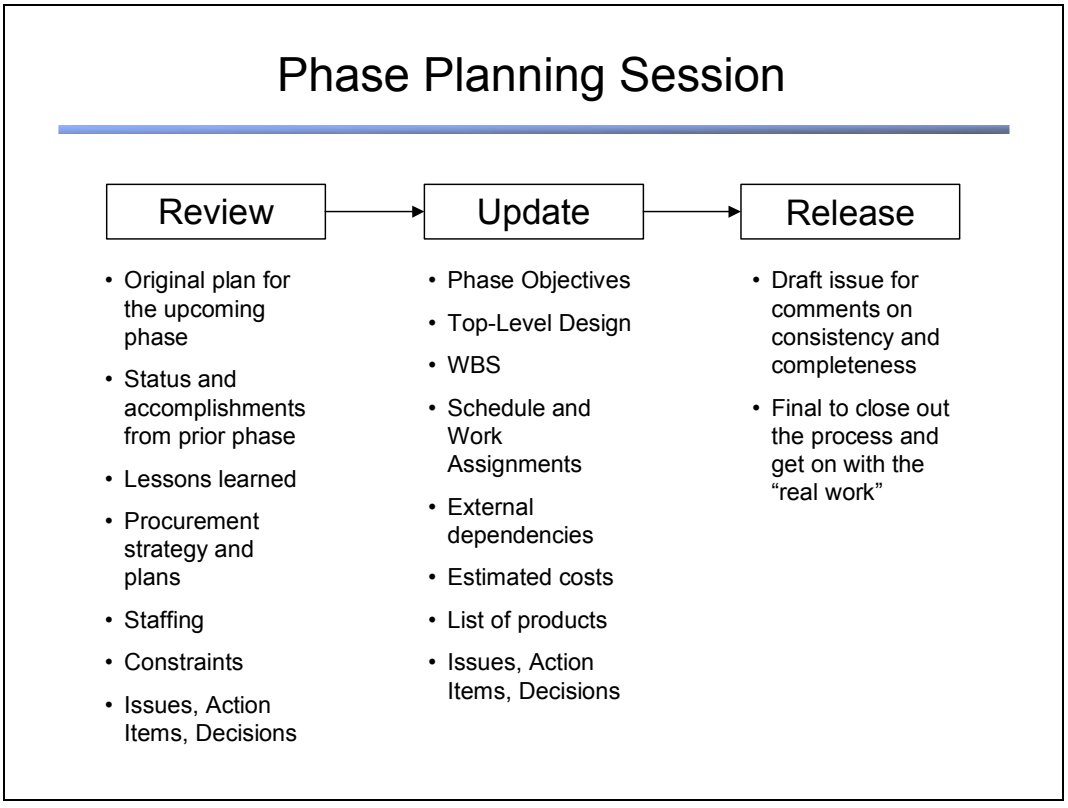


Figure E-1 Phase Planning Session Activity Breakdown

For the "review" and "release" stages, you'll probably want to keep the whole team together. For the "update" stage, you'll need to assess whether the work is best accomplished with the whole team together, or whether it would be more effective to break the group into smaller subgroups to discuss details and update their part of the plan.

Each of the activities is discussed in some detail below.

Review

As you think about what the upcoming phase entails, begin by reviewing your original plans and where you are now. Here's a list of what you might want to look at or think about:

- Original plan for the upcoming phase: Back at the beginning of the project, what did you think you would be doing in this phase? Are the COACH commitments, goals, builds, and assignments still valid?
- Status and accomplishments from the prior phase: As you wrap up the current phase, where are you? What did you accomplish? Was that more or less than you planned? How should the outcome of the current phase affect the next phase?
- Lessons learned: Study both your successes and shortcomings. Appendix F has some ideas about how to collect lessons learned.
- Procurement strategy and plans: What do you need to procure? What's the philosophy that guides how you will accomplish the procurements? Are there existing subcontracts that should be closed or renewed? What else?
- Staffing: Identify any problem areas or needs.
- Constraints: Are there any new regulations or external forces that should be accounted for as you update the plan for this phase?
- Issues, Action Items, Decisions: Revisit what you've said before; face the issues that affect this phase; make sure action items that involve the objectives for this phase have been closed.

Update

You may want to divide the team into smaller sub-teams to make the updates. A reminder list is given below:

- Phase objectives – Refine what you originally planned, including shifting any unmet objectives from previous phases.
- WBS – Extend and flesh out the lower-level details.
- Schedule and Work Assignments – Identify the start time (or criteria) and duration for every task in the phase. Make sure that every task in the phase is assigned to someone.
- External dependencies – Check the status of external deliveries or other external dependencies (specifications, data), and update the expected dates in your own schedule. Be sure that the linkages between external dependencies and project tasks are correct.

- Estimated costs – Re-figure the estimate to complete the phase, and make sure that you have adequate funds to keep the team going.
- List of products – Make sure that the tangible products of the phase are included in the WBS. Every major deliverable should be included as well as support products such as documentation. Some deliverables could also be highlighted as milestones for extra visibility.
- Issues, Action Items, and Decisions – Update the lists to show new assignments, closures, and progress or stuck-points, as shown in Figure E-2.

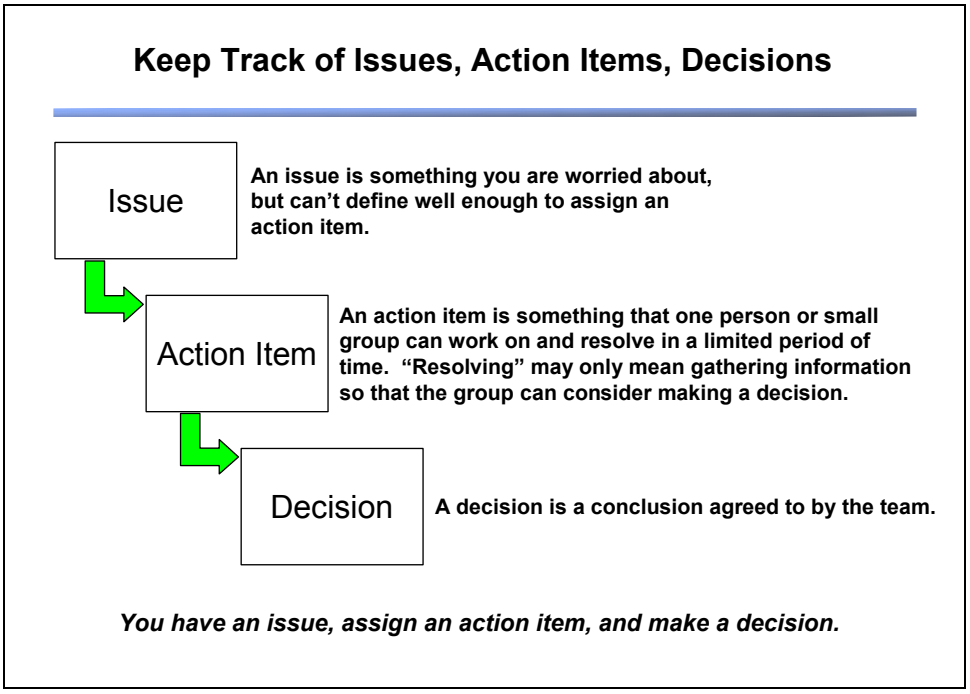


Figure E-2 Maintain an Informal Database of Issues, Action Items, Decisions

Release

Once the updates have been made, you should gather them together and make sure that there are no conflicts, and that everything is internally consistent.

In some phases you may have detailed information that you also want to summarize for external distribution. For instance, you may want to summarize the status of your interoperability testing for the benefit of an external audience.

If the phase coincides with a fiscal planning cycle, you may need to summarize past and estimated future financial information such as labor, travel, subcontracts, and other direct costs. Related to that would be your organization's need to formally enter anticipated budget needs for future fiscal years. It is not unusual to have a 2-year planning horizon for funding requirements.

The next section contains a sample agenda for a phase planning session. You should adjust the agenda topics and durations based on what makes sense and works well for your program or project.

E.3 Sample Agenda for Phase Planning Team Work Session

Day 1

8:00-8:30	Coffee Available
8:30-8:45	Introductions & Overview of Schedule
8:45-10:00	Review of Key Milestones, Planned versus Actual Activities, Issues
10:00-10:15	Break
10:15-10:45	Lessons Learned Session
10:45-11:15	Update on Credentials Administration Systems
11:15-11:30	Update on Safety Systems
11:20-12:00	Review Phase Plan Methodology & Contents
12:00-12:45	Working Lunch Provided
12:45-2:25	Update/Develop Phase Objectives, Implementation Steps, and Milestones (Breakouts)
2:25-2:40	Break
2:40-3:10	Report Back to Group
3:10-3:25	Lessons Learned Gathering and Multivoting (prioritizing)
3:25-3:30	Day 1 Wrap-up

Day 2

8:00 - 8:30	Coffee Available
8:30 - 8:35	Overview of Day's Schedule
8:35 - 8:45	Lessons Learned Results Presentation
8:45-10:00	Update System Design (Breakouts & Report Back to Group)
10:00 - 10:15	Break
10:15 - 12:00	Update WBS & Phase Products (Breakouts & Report Back to Group)
12:05 - 12:45	Working Lunch Provided
12:45 - 1:30	Update Phase Resources
1:30 - 2:30	Update Phase Schedule, External Dependencies, & Work Assignments
2:15 - 2:30	Break
2:30 - 3:00	Report Back to Group – Phase Schedule & External Dependencies
3:00 - 3:20	Report Back to Group – Work Assignments
3:20 - 3:30	Planning Session Wrap Up

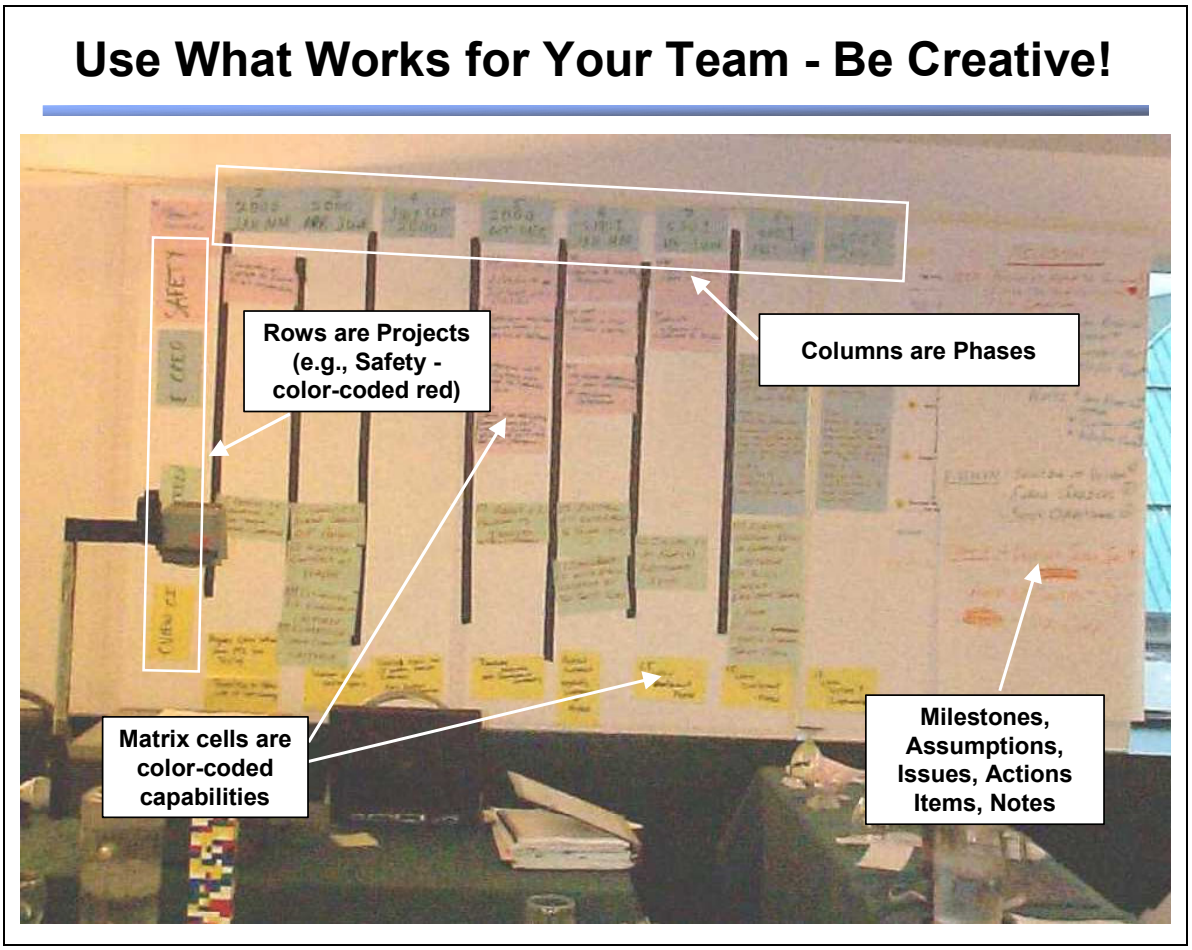


Figure E-3 Simple Tools Will Work Fine

Although there are many fine computer-assisted planning tools, simple tools, such as that shown in Figure E-3, work fine and allow the whole team to participate. That photo shows how some state teams at CVISN Planning Workshops used a white wall and construction paper to create this planning matrix for their breakout sessions.

APPENDIX F. HOW DO I CONDUCT A TEAM LESSONS LEARNED SESSION?

Cast a net for management lessons learned from time to time, especially as one phase is ending and another is to begin. A balanced process will gather both positive and negative perceptions and experiences of the team's work to date. An orderly and fair process will ensure that all participants have a chance to contribute. Below are ways to design a well-structured process so as to bring about those results.

F.1. Preparing for the Lessons Learned Session

- Give the group time in advance to collect their thoughts.
- Typical questions might be: "What worked well during this phase?" "What would you change if you were in charge?" Customize the topics to reflect what is appropriate in your organization and at this particular point in your CVISN work.
- Reserve a meeting room. Have blank flip charts and markers available. Post the topics for all to see.
- If you have a camera on hand take photographs of the resulting flip charts (one at a time). If you later scan the photographs (or used a digital camera) you will then have an electronic record that is easy to distribute via e-mail.

F.2. Guidelines for Structured Brainstorming

(These steps and precautions will ensure that the session remains under control and does not collapse into a chaotic free-for-all, nor be dominated by the person with the loudest voice.)

- Assign a facilitator and a scribe. For large groups (15 or more) use two scribes and two different flip charts to keep the process moving smoothly.
- Review the process guidelines with the participants.
- Review the topic questions to be worked on.
- Brainstorm items and record them on the flip chart paper.
- Enforce the round-robin method – one item per person at a time. The facilitator goes around the group again and again; each participant conveys one of their ideas per turn. No pressure – a person may say "pass".
- Free wheel; don't hold back any ideas even if they seem silly at the time.
- Minimize discussion during brainstorming! That comes later.
- Ensure that ideas are not evaluated when they are initially presented, but do give an opportunity for clarification and questions about an item before items are combined and the voting occurs.
- Hitchhike – build on the ideas of others.
- Suspend judgment – no one is allowed to criticize or applaud anyone's idea; not a groan or a grimace.
- Folks hold war stories to one minute please.
- Post the flip chart paper around the room as needed to accommodate everything.

The next section describes a process for consolidating and prioritizing the above free-form input, in order to produce a coherent prioritized summary.

F.3. Guidelines for Prioritizing and Multivoting

(These steps will ensure that a consensus-based written report emerges from the session.)

- Clarify as necessary. The person who contributed a puzzling item should initially explain the item but others may join in the discussion to help refine and focus the wording.
- Number each item from "a" through "z", "aa through "zz", …
- Ask the team to suggest grouping of related items.
- Check for agreement on grouping via an informal voting procedure: thumbs-up: "yes"; thumbs-down: "no"; thumbs-sideways: "can live with it". (Counts as "yes".) If one person disagrees, don't combine. Mark up the flip charts to group items agreed as similar.
- Finally, ask each member to vote priorities on the grouped items. If there are lots of items (more than about 10) then have everyone silently pick their top 25% and mark those (called "multivoting"). Like magic the high-priority items will receive the most checkmarks.
- Record the number of votes along side each item.
- After the lessons learned session, re-order the list by number of votes cast for each, and distribute the final list.

After the process is completed and results have been reported, the team may want to pick some specific items to work on in order to prevent or ensure their occurrence during the next phase.

Periodically, **share your lessons learned** with other states and FMCSA. There have been many instances of formal lessons learned reports; some are discussed in Chapter 7 of the *CVISN Guide to Program and Project Planning* [4] and cited in the references of that document.

For more background on structured brainstorming and multivoting, please see Reference [22].

APPENDIX G.SAMPLE ACTION ITEM, ISSUE, DECISION FORM

✓ ☐ ☐ ☐

Action Item / Issue / Decision

for

(event) _____

Date: _____

Originator: _____
(plus organization & phone number if new)

Topic: _____

Assigned to: _____

Due date if any: _____

Background & Context: _____

Description: _____

This Page Intentionally Blank