4. CVISN Planning Principles & Concepts

Management is what you do to get the right things done by other people. In this chapter, we list and discuss some principles on which the CVISN planning process is based. Then we discuss important planning concepts. Principles are "rules" to follow. Concepts are "abstract ideas" that are generalized from particular instances or experience and can be applied to new situations. The intent is that states will incorporate these principles and concepts into their own processes and procedures.

4.1 Planning Principles

The guiding principles for program and project planning are very similar. Those for program planning are related to the program's wider vision, and those for project planning reflect the project's narrower scope. Figure 4–1 lists the principles for program and project planning.

Remember that you don't need the perfect plan in place before you start making progress. Also remember the "Pareto Principle" which asserts that you get 80% of the effects from 20% of the causes! Be sure you leave time and energy to do "real work".

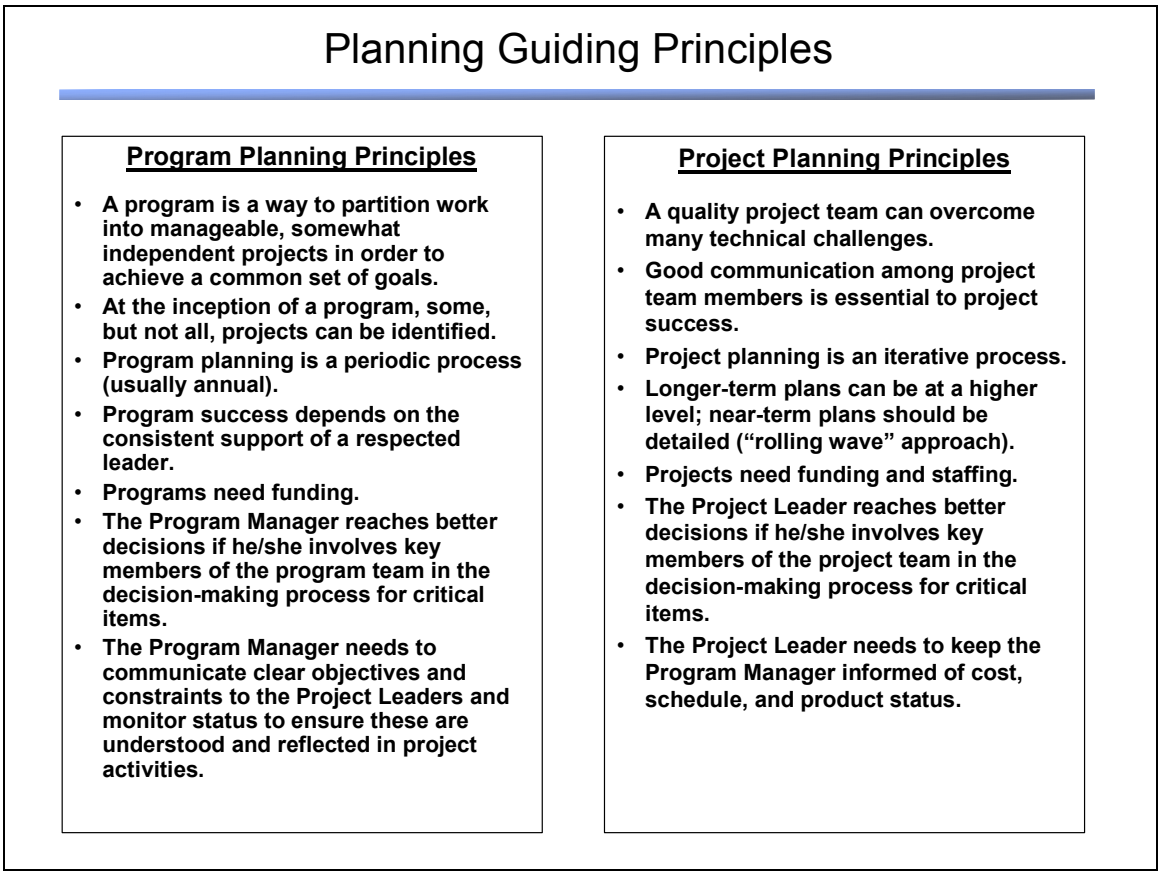

Figure 4-1. Planning Principles for Programs and Projects

4.1.1 Program Planning Guiding Principles

1. **A program is a way to partition work into manageable, somewhat independent projects in order to achieve a common set of goals.** Key here are two notions: that programs can be partitioned into separately-managed projects, and that those projects are partially dependent on each other. What ties the projects together are the common goals for the program, and the need for interoperability.

2. **At the inception of a program, some, but not all, projects can be identified.** As an Intelligent Transportation System program, CVISN is focused on improving safety and operations using technology. Technology changes rapidly. At the start of a CVISN Program in a state you can define the initial projects you want to undertake, but you need to be open to new opportunities such as when chances to make other improvements arise, and as technologies emerge and mature.

3. **Program planning is a periodic process (usually annual).** Program planning is typically tied to enterprise funding cycles. Most funding cycles are annual.

4. **Program success depends on the consistent support of a respected leader.** The program needs a champion who genuinely believes in the program's objectives, who can rally the project teams, and who can get the needed political support and funding. Success depends on the program manager's consistent support from both above and below.

5. **Programs need funding.** Programs can't accomplish anything without financial support. You always want to keep the funding stream one step ahead of the expense stream so that a project team doesn't get erratic direction, or worse, shut down for temporary lack of funding.

6. **The Program Manager reaches better decisions if he/she involves key members of the program team in the decision-making process for critical items.** Many heads are better than one. Involving key team members in the decision-making process not only results in better decisions, but also gives those folks a sense of ownership and self-respect. The Program Manager can't accomplish anything on their own. The projects get the work done.

7. **The Program Manager needs to communicate clear objectives and constraints to the Project Leaders, and monitor status to ensure these are understood and reflected in project activities.** The program's objectives must influence every project's objectives and goals. Every program operates under constraints such as schedule, cost, and resource availability. The Program Manager must communicate the wider objectives and real-life constraints to the Project Leaders.

4.1.2 Project Planning Guiding Principles

1. **A quality project team can overcome many technical challenges.** If you get quality people onto the project team, and they are committed to the project's success, they will overcome almost every technical challenge they face. There is nothing more important to a properly-funded project than having the right people on the team.

2. **Good communication among project team members is essential to project success.** There are many tools today to make it easier for team members to stay in touch. Use them. If everyone knows what is going on, it is easier to work together for the common goals.

3. **Project planning is an iterative process.** Projects get the work done. Keeping things on track is a continuous process of re-evaluating priorities, re-assigning resources, checking for and resolving problems, and keeping a few steps ahead of the current activities. This continuous planning and evaluating occurs as part of phase planning and tracking [44].

4. **Longer-term plans can be at a higher level; near-term plans should be detailed ("rolling wave" approach).** You need to know where you are headed (the longer-term plans) and how to get there. But to be sure you hit the milestones along the way you also need to know how the activities over the next few months must be interwoven to achieve the near-term targets.

5. **Projects need funding and staffing.** Projects can't do anything without staff. Staff won't be assigned unless funding is allocated.

6. **The Project Leader reaches better decisions if he/she involves key members of the project team in the decision-making process for critical items.** As at the program level, many heads are better than one. Involving key team members in the decision-making process not only results in better decisions, but it gives those folks a sense of ownership and self-respect. The project leader can't accomplish much on his or her own.

7. **The Project Leader needs to keep the Program Manager informed of cost, schedule, and product status.** Funding usually originates at the program level, and the Program Manager is ultimately responsible for all the projects. The Program Manager delegates responsibility to the Project Leaders, but must be kept informed about how the projects are progressing. Project Leaders should be able to report status honestly so that the Program Manager gets a clear picture of problem areas.

4.2 Planning Operational Concepts

The practical operational concepts that apply the above principles are very similar for both program and project planning. Those for program planning are related to the broader program-level activities, and those for project planning reflect the project's product-oriented focus. Figure 4–2 lists the concepts for program and project planning.

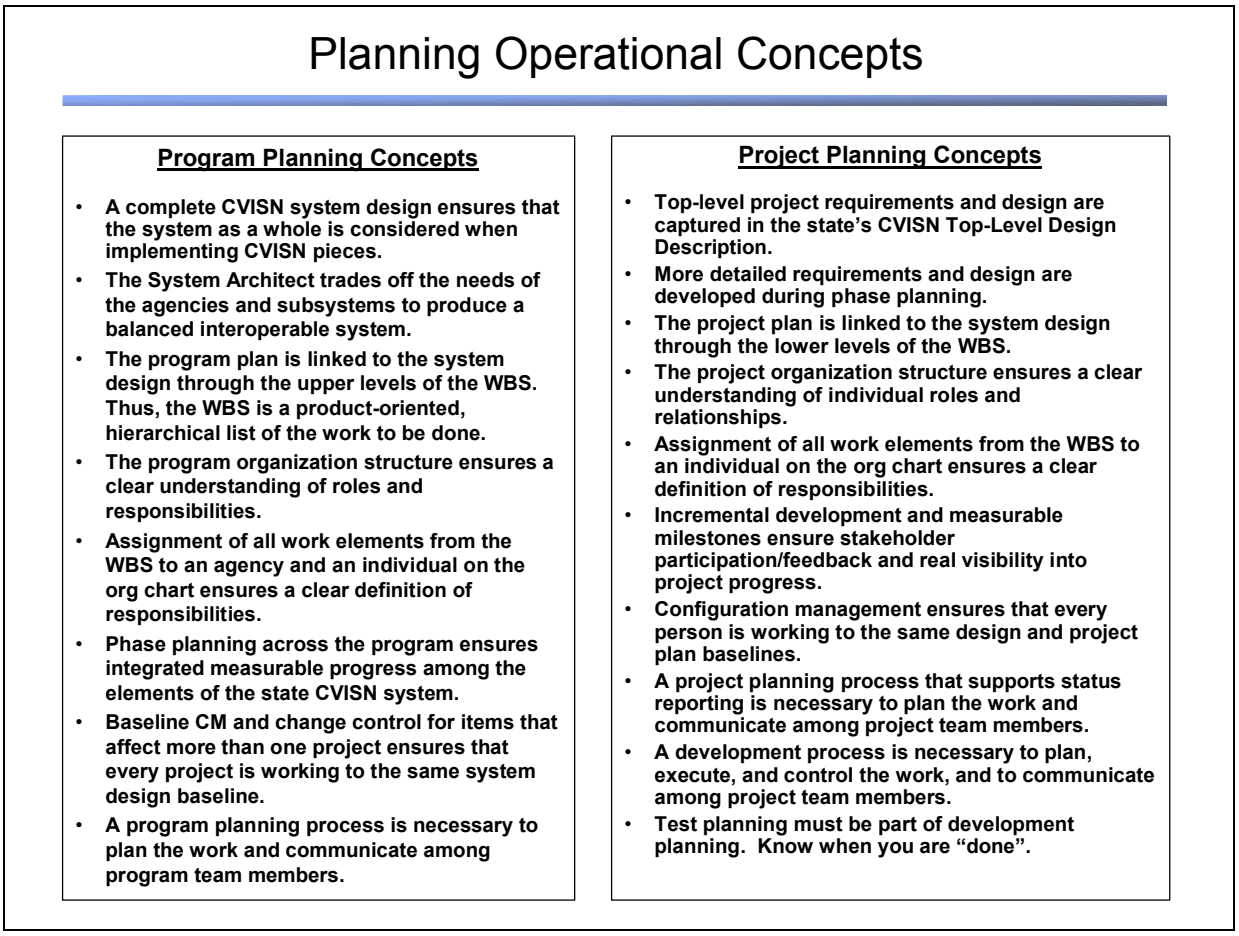

Figure 4-2. Planning Concepts for Programs and Projects

4.2.1 Program Planning Operational Concepts

1. **A complete CVISN system design ensures that the system as a whole is considered when implementing CVISN pieces.** During top-level design, the System Architect works with the technical leads of the project teams to establish a design that meets the program's and projects' objectives; defines end-to-end processes that make it possible to exchange information more effectively; and provides a common language and framework for discussing the CVISN projects. Your state's objectives and the system design define the scope of the CVISN Program and its resulting deployment projects. This is why the Scope Workshop precedes the Planning Workshop. Never lose sight of the system design!

2. **The System Architect trades off the needs of the agencies and subsystems to produce a balanced interoperable system.** As the top-level design emerges, it reflects the natural give-and-take of any group activity. The final design must be made robust enough to tolerate modest changes to requirements and the emergence of new technologies. The System Architect's job is to balance the demands from the project and the requirements associated with each system component, and then derive a system design that makes it possible for the parts to work together but be developed independently. A **balanced system** is a system in which no subsystem area is emphasized at the expense of the system as a whole. An **interoperable system** is a system in which all related systems and subsystems use common standards for data interchange and work together to accomplish shared processes.

3. **The program plan is linked to the system design through the upper levels of the WBS. Thus, the WBS is a product-oriented, hierarchical list of the work to be done.** If you could do only one planning task, it should be the Work Breakdown Structure. The classic definition of the WBS is **"a hierarchical breakdown of goods and services"** – goods in the sense of deliverable products; services in the sense of level-of-effort tasks such as system engineering and quality assurance. Figure 4–3 illustrates how the WBS links the system design to the program plan.

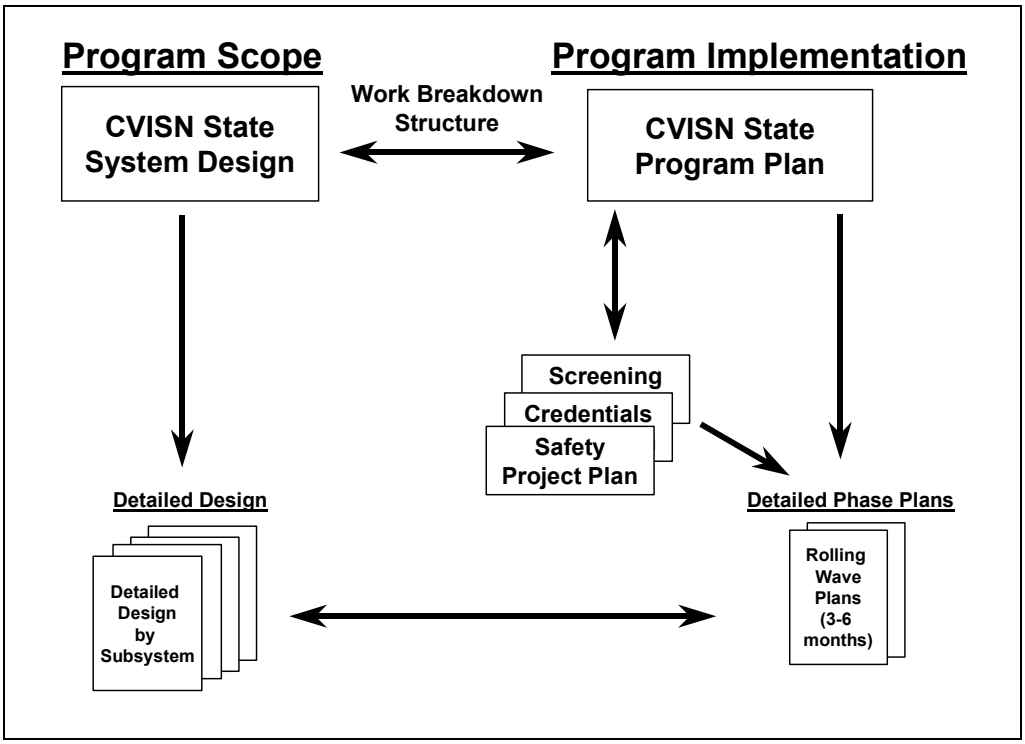

Figure 4-3. The Program Plan is Linked to the System Design Through the WBS

4. **The program organization structure ensures a clear understanding of roles and responsibilities.** The program organization structure is typically displayed using an organization chart in the form of an inverted tree. It is intended to be completely distinct from the state agencies' permanent reporting structure. The chart shows who is responsible for each project in the program, and what program-wide infrastructure support is needed. Please see chapter 5 for an example.

5. **Assignment of all work elements from the WBS to an agency and an individual on the org chart ensures a clear definition of responsibilities.** Mapping the top levels of the work breakdown structure to the agencies represented on the program organization chart captures which agency is responsible for each aspect of the program. Delegating responsibility helps to gain commitment to the program.

6. **Phase planning across the program ensures integrated measurable progress among the elements of the state CVISN system.** Coordinate the development schedules across the projects by grouping tasks into phases to achieve early benefits from the CVISN Program. For example, if one part of the system provides needed data to another part, everyone understands what the program can accomplish as a whole. Reference [37] has many heuristics for deciding what kind of tasks to combine and what to keep separate.

   Each planning level (program, project, and product) may have its own independent phases. Figure 4–4 illustrates this concept.

Phases and Deliveries

A "phase" is a management convenience – a period of calendar time specified for planning purposes to allow incremental delivery of a complex system. A "phase" is not a WBS element, but rather a portrayal of how the WBS elements are developed over time.

Each planning level (program, project, product) can have its own independently-established phases.

In a . . . ⇩	A Phase is Called	What is Being "Delivered" Is Called
Program	Program Phase	Integrated Capabilities
Project	Project Phase	A Build
Product	Product Phase	Version or Deployment

Each "build" is an operational version of a system that incorporates a specified subset of the capabilities that will ultimately be provided.

A product's "version" identifies its particular configuration at the time of distribution.

Product "deployment" means the installation of a version for customer use. Typically not every version is deployed; some are for internal use only.

Figure 4-4. Development Phases and Product Deliveries

In the software development world "build" is not only a verb but also a noun used to describe a software entity. We use "build" to mean an integration of particular versions of products into a working system.

Phase planning involves two key components: establishing the schedule for each phase, and defining the objectives for each phase. Figure 4–5 illustrates a sample program phase chart. Phases need not be the same length, but are typically 3–6 months long. They do not overlap.

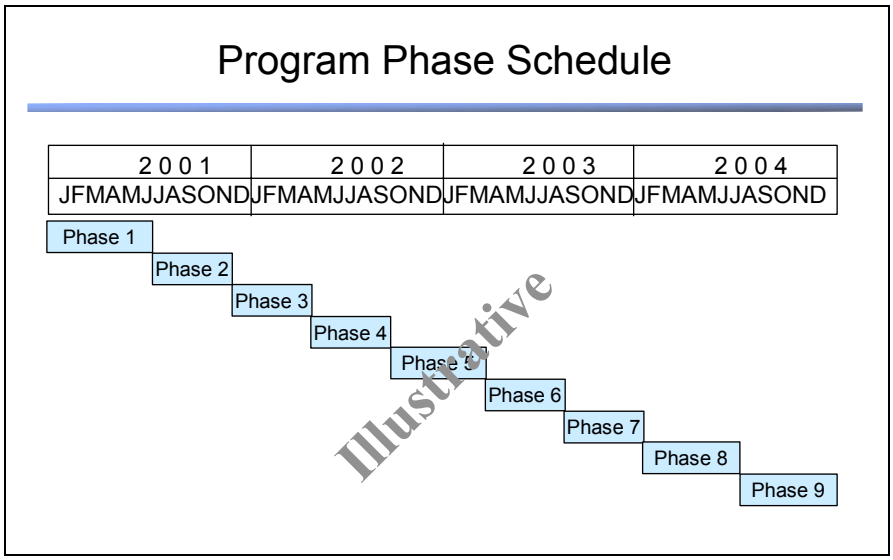

Figure 4-5. Illustrative Program-Wide Phase Chart

Figure 4–6 shows sample objectives for one phase. A similar list of objectives would be developed for each phase.

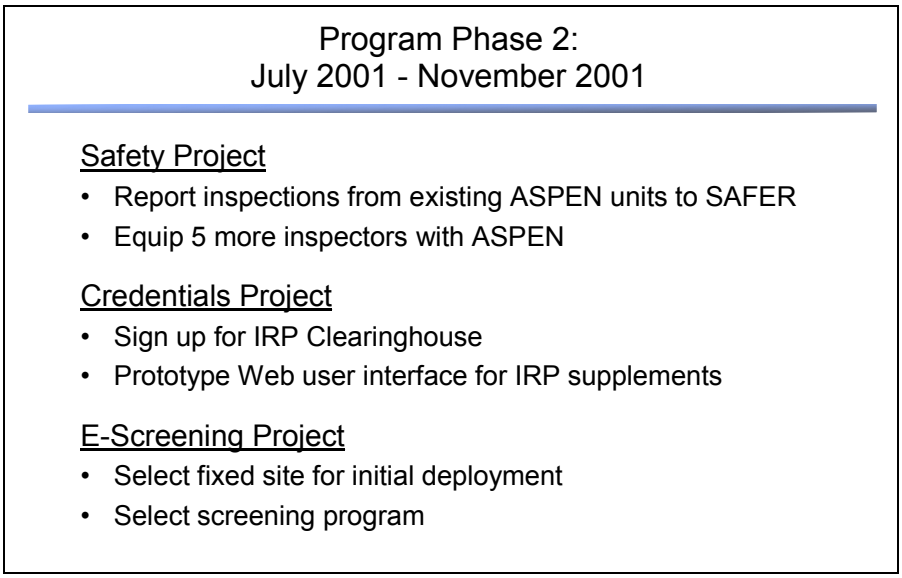

Figure 4-6. The Definition of Program-Wide Phases Encourages Incremental Integration and Interoperability Across Projects

7. Baseline configuration management and change control for items that affect more than one project ensures that every project is working to the same system design baseline. Configuration management is a discipline for managing the evolution of the program's products. Since many products are being developed incrementally and in parallel, you must have a way to keep track of requests for changes to requirements and how they might impact the design, costs, and schedules. You should establish a program-level configuration control board (CCB) chaired by the System Architect and consisting of technical representatives from all the projects to manage change requests that impact more than one project.

8. **A program planning process is necessary to plan the work and communicate among program team members.** The process of planning the program builds the program team; opens the lines of communications across projects; and defines a common set of expectations. When the CVISN Program involves multiple agencies, the open nature of the planning process should also improve communications at managerial levels among those agencies.

4.2.2 Project Planning Operational Concepts

1. **Top-level project requirements and design are captured in the state's CVISN Top-Level Design Description.** By putting the top-level requirements and design in the state's CVISN Top-Level Design Description, projects are assured of a baseline top-level design that is consistent and handles everyone's high-level requirements.

2. **More detailed requirements and design are developed during phase planning.** In the recommended spiral development model (described in the *CVISN Guide to Phase Planning and Tracking* [44]), the system is deployed incrementally by successive iterations through design, build, 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 this 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.

3. **The project plan is linked to the system design through the lower levels of the WBS.** The WBS is the vehicle to divide work into tasks that are:

- Manageable – Team leaders and staff can direct themselves within the confines of technical, cost, and schedule goals and limitations.
- Independent – Self-guided development teams perform with the peace of mind that well-defined interfaces were created via the system engineering skills of the System Architect, so they need not worry day-to-day what other teams are doing.

- Measurable – Technical specifications and acceptance criteria are available so that developers know that they are on target.
- Integratable – Elements come together physically and functionally to form a working system.

4. **The project organization structure ensures a clear understanding of individual roles and relationships.** The project organization structure is typically portrayed in the form of an inverted tree. (It is intended to be completely distinct from the state agencies' permanent reporting structure.) The chart shows the task leaders in the project context. Please see chapter 5 for an example.

5. **Assignment of all work elements from the WBS to an individual on the organization chart ensures a clear definition of responsibilities.** Work gets done when real individuals are assigned specific tasks. WBS element task leader will likely delegate execution of lower-level activities to a specific team member. The assignment process includes the development of detailed schedules and commitment of resources.

6. **Incremental development and measurable milestones ensure stakeholder participation/feedback and real visibility into project progress.** With adequate funding and proper coordination, work products will unfold. Identifying critical milestones in each phase helps you measure whether the phase is on track or not. Figure 4–7 is a high-level view in which milestones are like teeth in the gears of the "program machine".

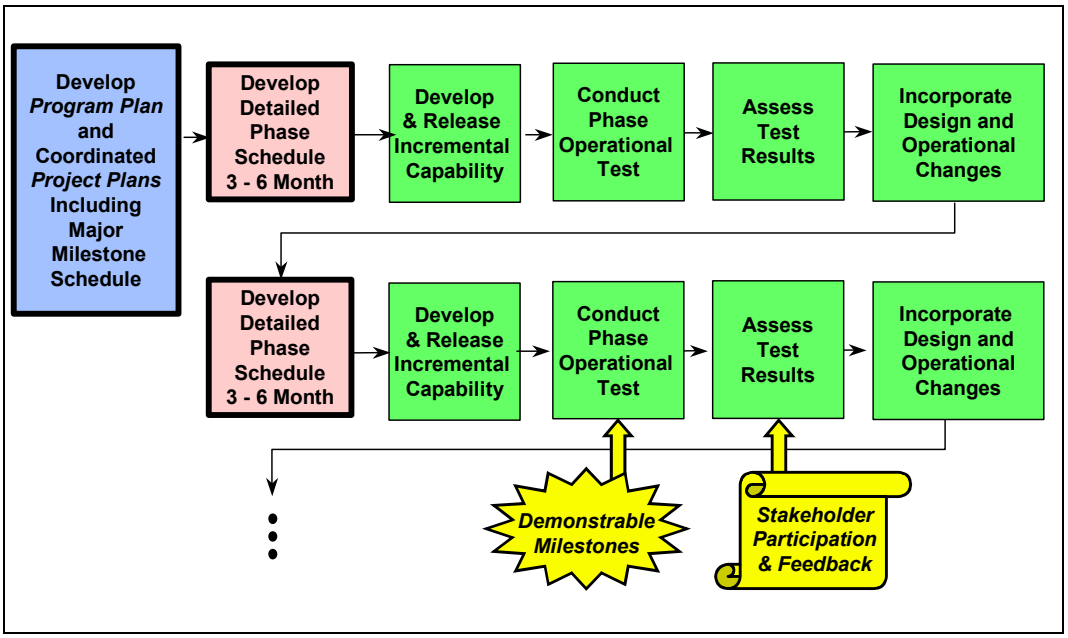

Figure 4-7. Demonstrable Milestones in Each Phase

7. **Configuration management ensures that every person is working to the same design and project plan baselines.** 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, when, and where proposed resolutions are implemented.

   Figure 4–8 shows the relationships among phases, builds, and product versions. Each program phase is associated with a set of defined operational builds within each functional area. Each project's build is a set of specific versions of products ("releases") that have been integrated and tested together, to provide specified capabilities in a functional area. Every product release gets a version number and associated description. All of the preceding are under configuration management.

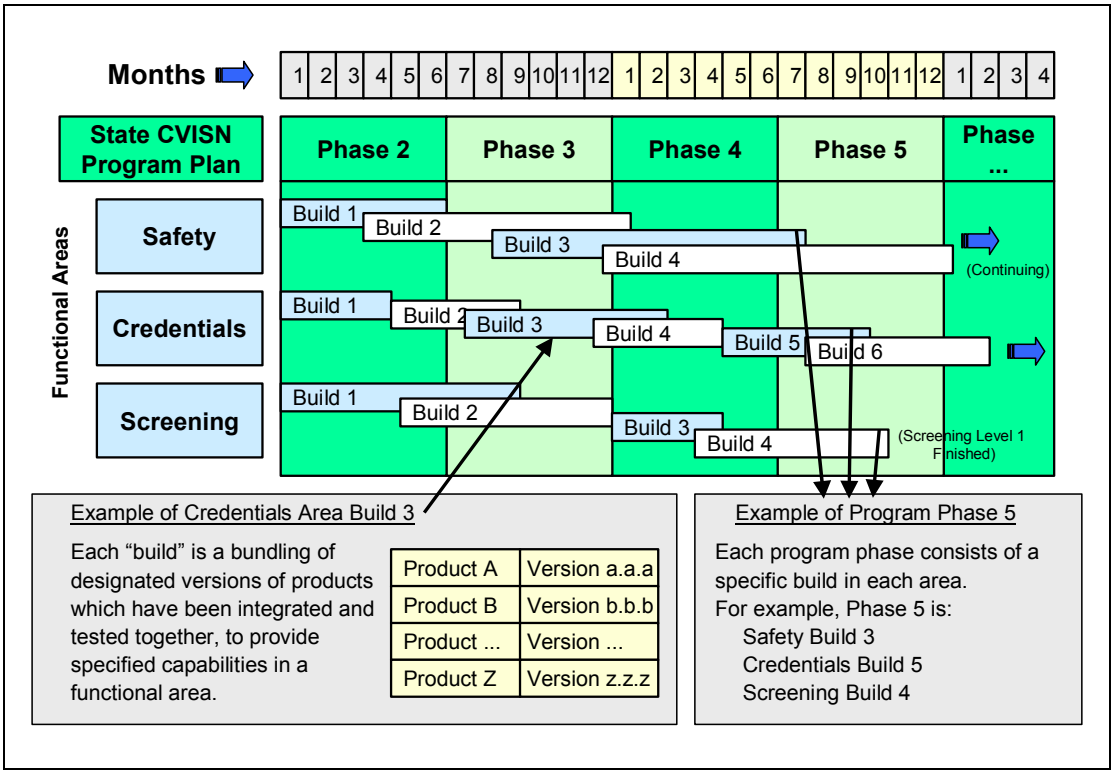

Figure 4-8. Relationship Among Phases, Builds, and Product Versions

8. **A project planning process that supports status reporting is necessary to plan the work and communicate among project team members.** The project team will be focused on getting work done. Interrupting them to collect status should be minimized. If both the planning and the status reporting processes can be integrated so that planning vehicles are updated to report status, both processes will be as painless as possible. The *CVISN Guide to Phase Planning and Tracking* [44] discusses this in more detail with examples.

9. **A development process is necessary to plan, execute, and control the work, and to communicate among project team members.** Product development is a team effort. So that each member of the team can work effectively, you need to put in place clear and simple procedures and processes for how the team will accomplish the development. For instance, there should be some common scheme for identifying and describing versions of software. Figure 4–9 shows how versions might be related to builds for phases.

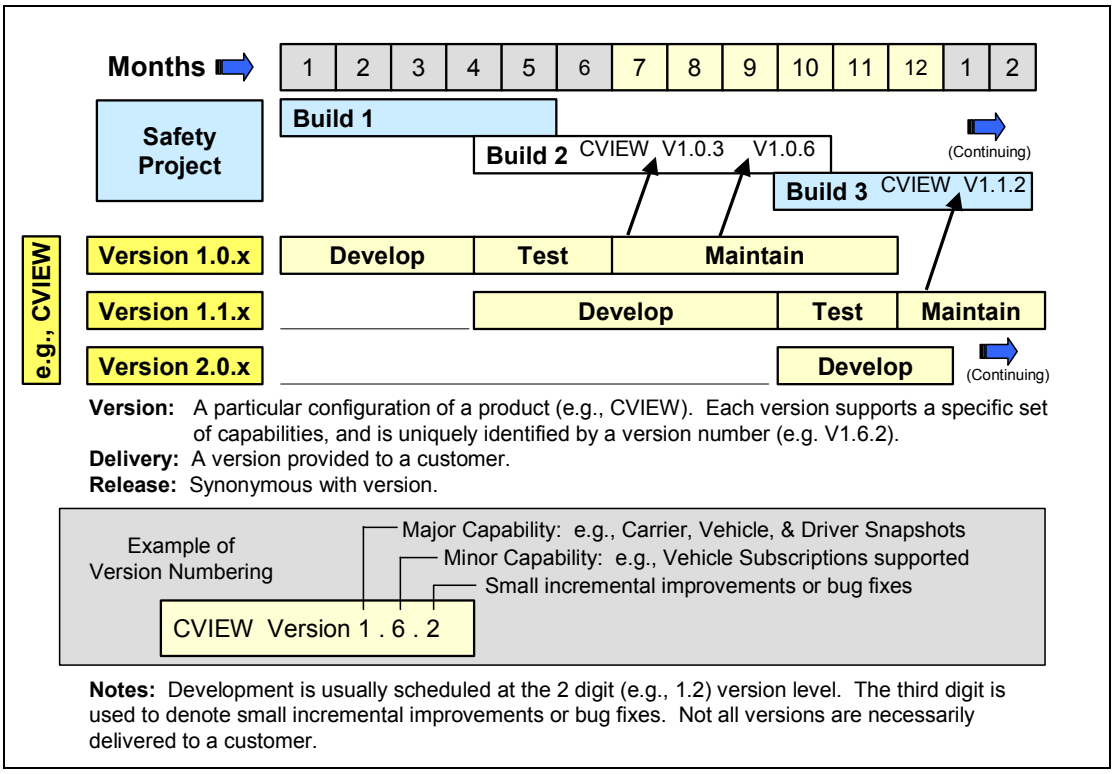

Figure 4-9. Version Identification is One Typical Development Process

Figure 4–9 depicts that at any point in time, different products are likely to be in different stages of their development lifecycle. Other development processes should also be nailed down so that everyone can communicate and get their own part of the whole job done effectively.

10. **Test planning must be part of development planning. Knowing when you are "done" is important!** Testing is the way you demonstrate that you have actually accomplished the system development you set out to do. To have a successful demonstration though, the test criteria must mesh with the project requirements; test procedures must reflect how the system actually operates in production use; and realistic test data sets must be available. The system must be designed for testability in the first place.