Elements of Software Configuration Management

Software configuration management (SCM) is one of the disciplines of the 1980's which grew in response to the many failures of the software industry throughout the 1970's. Over the last ten years, computers have been applied to the solution of so many complex problems that our ability to manage these applications has all too frequently failed. This has resulted in the development of a series of "new" disciplines intended to help control the software process.

product integrity

The goal of the software developer is, or should be, the construction of a product which closely matches the real needs of the set of people for whom the software is developed. We call this goal the achievement of "product integrity." More formally stated, product integrity is defined to be the intrinsic set of attributes that characterize a product:

* that fulfills user functional needs.
* that can easily and completely be traced through its life cycle.
* that meets specified performance criteria.
* whose cost expectations are met.
* whose delivery expectations are met.

Now that we more fully appreciate what we are after, i.e., to build a software product with integrity, let us look at the one road which might get us there. We have, until now, used the term "developer" to characterize the organizational unit responsible for converting the software idea into a software product. But developers are a complex set of interacting organizational entities. When undertaking a software project, most developers structure themselves into three basic discipline sets which include:

* project management
* development
* product assurance

The development disciplines represent those traditionally applied to a software project. They include:

* analysis
* design
* engineering
* production (coding)
* test (unit/subsystem)
* installation
* documentation
* training
* maintenance

The product assurance disciplines which are used by project management to gain visibility into the development process include:

* configuration management
* quality assurance
* validation and verification
* test and evaluation

**THE ELEMENTS OF SCM**

* identification
* control
* auditing
* status accounting

**ROLE OF SOFTWARE CONFIGURATION**

One role of software configuration control is to provide the administrative mechanism for precipitating, preparing, evaluating, and approving or disapproving all change proposals throughout the system life cycle. For that involves three basic ingredients.

1) Documentation (such as administrative forms and supporting technical and administrative material) for formally precipitating and defining a proposed change to a software system.

2) An organizational body for formally evaluating and approving or disapproving a proposed change to a software system (the Configuration Control Board).

3) Procedures for controlling changes to a software system. The Engineering Change Proposal (ECP), a major control document, contains information such as a description of the proposed change, identification of the originating organization, rationale for the change, identification of affected baselines and SCI's (if appropriate), and specification of cost and schedule impacts. ECP's are reviewed and coordinated by the CCB, which is typically a body representing all organizational units which have a vested interest in proposed changes.

A Program Support Libraries (PSL) should support three main activities: code development, software management, and configuration control. Support to the development process includes support to design, coding, testing, documentation, and program maintenance along with associated database schema and subschema. A PSL provides this support through:

* storage and maintenance of software documentation
* support to program compilation/testing,
* support for the generation of program/system documentation.

Support to the management of the software development process involves the storage and output of programming data such as:

* collection and automatic reporting of management data related to program development,
* control over the integrity and security of the data in the PSL,
* separation of the clerical activity related to the programming process.

PSL's provide support to the configuration control process through:

* access and change authorization control for all data in the library,
* control of software code releases,
* automatic program and document reconstruction,
* automatic change tracking and reporting,
* assurance of the consistency between documentation, code, and listings.

Although administrative in nature, status accounting is a function that increases in complexity as the system life cycle progresses because of the multiple software representations that emerge with later baselines. This complexity generally results in large amounts of data to be recorded and reported. In particular, the scope of software configuration status accounting encompasses the recording and reporting of:

1) the time at which each representation of a baseline and update came into being;

2) the time at which each software configuration item came into being;

3) descriptive information about each SCI;

4) engineering change proposal status (approved, disapproved, awaiting action);

5) descriptive information about each ECP;

6) change status;

7) descriptive information about each change;

8) status of technical and administrative documentation associated with a baseline or update (such as a plan prescribing tests to be performed on a baseline for updating

purposes);

9) deficiencies in a to-be-established baseline uncoveredduring a configuration audit.