Software Configuration Management Plan

Introduction

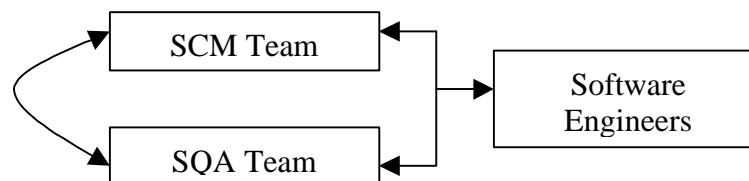## Scope and Intent of SCM Activities

The primary focus of the Software Configuration Management (SCM) is to identify and control major software changes, ensure that change is being properly implemented, and report changes to any other personnel or clients who may have an interest.

The objective of SCM is to limit the impact changes may have on the entire system. This will help to eliminate unnecessary changes, and to monitor and control any necessary changes. This allows software development to continue, despite large and/or insignificant changes without significant backtracking, lessening development time and resulting in a higher-quality product.

The SCM team will oversee these activities, and any changes to existing code or architectural design must pass their inspection before they are carried out.

SCM Organizational Role

The SCM team will work closely with the SQA (Software Quality Assurance) team, cross-examining many of the submitted documents and software change requests. Software Engineers will submit change requests directly to the SCM team for their inspection and approval.

```
   ┌──────────────┐
   │  SCM Team    │◄─┐
   └──────────────┘  │   ┌──────────────┐
                     ├──►│  Software    │
   ┌──────────────┐  │   │  Engineers   │
   │  SQA Team    │◄─┘   └──────────────┘
   └──────────────┘
```

An SCM leader will be appointed to oversee all SCM activities. He will receive all change requests, and will make any final decisions regarding those changes, including which software engineer will carry out approved changes. The SCM leader also keeps a library of all submitted requests, even those that have been denied.

SCM Tasks

Identification

Description

The SCM leader will analyze all current design specifications and break down the software into subsystems. All subsystems will consist of major software functions or interface components. Any submitted changes will be connected to its corresponding subsystem, which will be traced backwards though the system to determine its impact.

Work Products

An SCI document will contain the breakdown of subsystems and how they are interrelated. Any approved changes will be returned to the software engineer with an change approval sheet, a listing of all possible affected subsystems, and any additional information the software engineer may need before he begins changing code.

Configuration Control

Description

Software engineers will submit a change request to the SCM leader. The SCM leader will then analyze the request, using the SCI document, the project design document, and the current prototype of the software. He will base his decision on how severely the change will impact the entire system and, more importantly, on the corresponding subsystem.

Once his decision has been made, he must submit the change to the software engineer of his choice, as well as updating the SCI document to accommodate the change, and the SCM library to record the change request and decision.
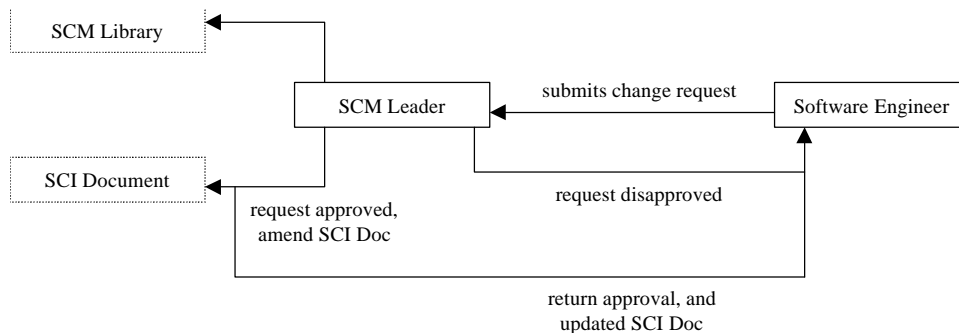
Description of Configuration Control Task: **Submitting Change Requests**

> Software engineers will submit a change request to the SCM leader. Because PA Software is a very small development house, all personnel work closely with each other, resulting in a more informal atmosphere. Because of this, formal request documents are not required of the software engineer when he submits change requests. These requests may be done via email or by word of mouth (so long as the SCM leader documents the request). Formal requests will always be accepted, but PA Software does not retain a standard form. All requests, whether they are approved or not, are recorded in the SCM Request Library. Records include the requesting engineer's name, the date of the request, the subject of the request, and eventually whether or not it was approved.

Critique: This approach is reasonable in this context. However, we need to know who the "SCM leader" is by name and what other responsibilities this person has. Key here is to ensure that SCM gets done and that someone is responsible for making sure that it does get done.

Description of Configuration Control Task: **Request Analysis**

> The SCM leader will then analyze the request, using the SCI document, the project design document, and the current prototype of the software. The SCI document will be used to track the impact through the corresponding subsystem, and eventually the entire system. The original design document will be consulted to ensure that the requested change remains within requirement specifications and the over all spirit of the project. The SCM leader may consult the current prototype if he is unsure any change is required. This is imperative for cosmetic changes to the interface. He will base his decision on how severely the change will impact the entire system and, more importantly, on the corresponding subsystem.

Description of Configuration Control Task: **Request Disapproval**

> If the SCM leader deems the change unnecessary, he will contact the software engineer who made the request and explain the reason the request was denied. The software engineer may discuss this decision with the SCM leader at this time, and resubmit a modified change request if a different understanding is reached.

Description of Configuration Control Task: **Request Approval**

> If the SCM leader deems the change necessary, he will update the SCI document to reflect the change. This may include changes with the corresponding subsystem, and any impact it may have on the entire system. Once the SCI document has been amended, it will be returned to

the software engineer, and he will be notified of all possible affected subsystems for surveillance after the change has been introduced.

If these changes result in a drastic alteration of the software's functionality or the way in which users will interact with the software, the client will be contacted for approval prior the delivery of the approval to the software engineer.

Work Products and Documentation

Submitting Change Requests
SCM Library is updated to reflect the request.

Request Analysis
None.

Request Disapproval
SCM Library is updated to reflect the disapproval.

Request Approval
SCM Library is updated to reflect the approval.
The SCI is amended.

Version Control

Description

Whenever the system or a subsystem is updated, the program build number (version number) will be updated to reflect the change. The version number will follow a standard x.x.x input mask (for example, version 1.2.7), with each digit place corresponding to an increasing severity of change. The hundredths place (x.x.**x**) will reflect very minor changes to the software. The tenths place (x.**x**.x) will reflect more substantial software changes. The ones place (**x**.x.x) will reflect severe changes to the software. Version control will be done by hand. No source code tracking / version control tools will be used.

Critique: Manual control is OK in this context, but for industry projects, an automated system for version control is highly recommended.

Description of Version Control Task: **Minor Version Change**

When a minor bug is fixed in the software, the hundredths place digit will increase. This change reflects a singular error being corrected, or a minor design change that has little or no impact on the surrounding subsystem. These changes are usually singular, and may be numerous enough to affect

4

the thousandth place digit (for example, a tenth bug has been found and corrected during the current version of software, the version number would increase accordingly: 1.2.10)

Critique: Who is responsible for managing and tracking the version control number? Is there a central record of each SCI and its version? If so, who maintains it? If not, how can we be sure that we're building software at the same version level? These questions should be answered here.

Description of Version Control Task: **Substantial Version Change**

When a more substantial change is made to the software, the tenths place digit will increase. This change reflects the correction of many minor bugs simultaneously, or design changes that are substantial enough to affect the way the software operates internally or slightly alter the way the user interacts with the interface.

Description of Version Control Task: **Severe Version Change**

When a severe change is made to the software, the ones place digit will increase. This change reflects any design decisions that will affect the manor in which a significant portion of the subsystems interact with each other, a major change in the functionality of the software, or any interface overhauls severe enough to completely alter the way the user interacts with the interface.

Work Products and Documentation

Minor Version Change
The SCM library will note the version change and the bug that was fixed.

Substantial Version Change
The SCM library will note the version change and the bugs that were fixed or design changes were made. If design changes are made, the SCI document will also be updated.

Severe Version Change
The SCM library will note the version change and the design decisions that were made. All affected subsystems will be updated in the SCI document.

Configuration Status Accounting (CSA)

Description

Once the SCI document has been amended, it will be returned to the software engineer, and he will be notified of all possible affected subsystems for surveillance after the change has been introduced by the SCM leader, either by word of mouth or via email.

If these changes result in a drastic alteration of the software's functionality or the way in which users will interact with the software, the client will be contacted prior to any changes being made. A draft of the changes will be sent to the client for review. If the client deems the change as necessary, they may contact the SCM leader with their approval.

Work Products and Documentation

A draft of the changes will be sent to the client. Upon approval, the final change document will be amended to the SCI document.

Audits and Reviews

Description

Software configuration audits and formal technical reviews are held to ensure that change has been properly implemented. The SCM team is held to a standard process set by PA Software. Periodic reviews of the SCM teams actions determine if they are conducting their activities properly and thoroughly. This is done to make sure any changes that are made retain the original quality of the design, and do not introduce new defects or design flaws. The SCM leader will be contacted if any infractions are found. It will be his duty to take corrective action to correct the problem.

Work Products and Documentation

A review document will be produced, cataloguing the SCM team's performance, noting any problems they are having conforming to the standard processes.

Data Collection and Evaluation

All change requests are sent the SCM leader, who documents them and files them in the SCM library. A hard copy is kept in the SCM office, as well as electronic copies on the PA Software network.

Following PA Software's informal atmosphere, all but the most important information may be submitted informally, via email or word of mouth. All contact with clients may also be informal, unless the situation requires otherwise.

Standards, Practices and Conventions (SPC)

All software engineers are responsible for submitting any major design changes or implementation variances to the SCM leader. This procedure will account for all impacts on the rest of the software resulting from the change(s). Every major change will be recorded by the SCM leader, including which software engineer requested the change and the date requested.

All software engineers are expected to thoroughly test each completed subsystem of the software following guidelines outlined in the Test Specification. This is to ensure that each subsystem is working properly and efficiently. Any major defect found will be reported to the SCM leader.

All SCM team members will loosely follow a set of criteria when conducting their activities. A list of these guidelines is included in the appendix.

SCM Resources

Personnel
SQA team leader
SQA team members (3)

Hardware
No special hardware is required.

Software
No special software is required, but access to a central database containing the SQA defect log would be preferable in communicating between the SQA team and software engineers.

Tools
No special tools are required.

Software Quality Assurance Overview

The SQA team's objective is to insure that the product does not deviate far from the original design specifications. If it is discovered that deviation has occurred, the SQA team will notify the development team to prevent future deviations and to correct the previous deviations. Also, the SQA team will perform a walkthrough to analyze the product's quality at any particular stage of development. Error detection and possible enhancements are also expressed to the development team.

The SQA organizational role is to review the product(s) at specific times during product implementation. Upon reviewing, the SQA team's duties will be to evaluate the software at its current development stage and recognize any defects in the subsequent stage (design or implementation). The SQA team will directly interact with the software engineering team in group discussions, discussing any errors or possible enhancements that have been identified. In addition, the SQA team will ensure that the software engineering team has not deviated in any way from the initial design specifications.

SCM Tools, Techniques, and Methods

All SCM activities will follow the same guidelines and methods (see appendix). Every SCM meeting will include every group member. Every group member is expected to participate in the discussion. Any group member not attending the review will be notified by the SCM leader of what took place at the review. The SCM leader will oversee the discussion and will take notes of any defects or enhancements that need to be analyzed.

The SCM team will analyze the defects or enhancements and determine their complexity, impact on the system, and priority. Once prioritized, the SCM leader will assign each item to the software engineers along with their priority.

After a defect has been eliminated or an enhancement added, the software engineer will inform the SCM leader at the next SCM review. The SCM leader will take note of the correction in the SCM library.

No special tools will be necessary for SCM team although access to a central database that all group members can access would be preferable to cut down on time and duplication of errors.

Appendix

SCM Process Criteria

1. Has the changes specified in the SCM library been made? Have any additional modifications been incorporated?
2. Has a formal technical review been conducted to assess technical correctness?
3. Have software-engineering standards been properly followed?
4. Has the change been "highlighted" in the SCI? Have the change date and change author been specified? Do the attributes of the configuration item reflect the change?
5. Have the SCM procedures for nothing the change, recording it, and reporting it been followed?
6. Have all related SCIs been properly updated?