**Introduction to Computing and software**

**Engineering**

**Assignment-1**

**Group work**

**Prepared by:-**

1. **Biruk Mammo………………………………………ETS**
2. **Biruk Fekade………………………………………. ETS**
3. **Dawit Tegegn……………………………………… ETS**
4. **Endriyas Yeshidniber…………………………….. ETS**
5. **Tsion Tadele………………………………………..ETS**

**Dec – 2008 E.C**

1. ***Write a compiled short note about software configuration management?***

Software Configuration management (SCM) is concerned with the policies, processes, and tools for managing changing software systems. You need to manage evolving systems because it is easy to lose track of what changes and component versions have been incorporated into each system version. Versions implement proposals for change, corrections of faults, and adaptations for different hardware and operating systems. There may be several versions under development and in use at the same time. If you don’t have effective configuration management procedures in place, you may waste effort modifying the wrong version of a system, deliver the wrong version of a system to customers, or forget where the software source code for a particular version of the system or component is stored.

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 configuration management of a software system product involves four closely related activities

1. *Change management* this involves keeping track of requests for changes to the software from customers and developers, working out the costs and impact of making these changes, and deciding if and when the changes should be implemented.

2. *Version management* this involves keeping track of the multiple versions of system components and ensuring that changes made to components by different developers do not interfere with each other.

3. *System building* this is the process of assembling program components, data, and libraries, and then compiling and linking these to create an executable system.

4. *Release management* this involves preparing software for external release and keeping track of the system versions that have been released for customer use.

**SCM major practices**

When implementing SCM tools and processes, you must define what practices and policies to employ to avoid common configuration problems and maximize team productivity. Many years of practical experience have shown that the following best practices are essential to successful software development:

• Identify and store artifacts in a secure repository.

• Control and audit changes to artifacts.

• Organize versioned artifacts into versioned components.

• Organize versioned components and subsystems into versioned subsystems.

• Create baselines at project milestones.

• Record and track requests for change.

• Organize and integrate consistent sets of versions using activities.

• Maintain stable and consistent workspaces.

• Support concurrent changes to artifacts and components.

• Integrate early and often.

• Ensure reproducibility of software builds.

Let us define configuration management of software system product activities briefly,

1. **Change management**

Change is a fact of life for large software systems. Organizational needs and requirements change during the lifetime of a system, bugs have to be repaired, and systems have to adapt to changes in their environment. To ensure that the changes are applied to the system in a controlled way, you need a set of tool-supported, change management processes. Change management is intended to ensure that the evolution of the system is a managed process and that priority is given to the most urgent and cost-effective changes.

1. **Version management**

Version management (VM) is the process of keeping track of different versions of software components or configuration items and the systems in which these components are used. It also involves ensuring that changes made by different developers to these versions do not interfere with each other. You can, therefore, think of version management as the process of managing code-lines and baselines.

A code-line is a sequence of versions of source code with later versions in the sequence derived from earlier versions. Code-lines normally apply to components of systems so that there are different versions of each component. A baseline is a definition of a specific system. The baseline therefore specifies the component versions those are included in the system plus a specification of the libraries used, configuration files.

1. **System building**

System building is the process of creating a complete, executable system by compiling and linking the system components, external libraries, configuration files, etc. System building tools and version management tools must communicate as the build process involves checking out component versions from the repository managed by the version management system. The configuration description used to identify a baseline is also used by the system building tool.

1. **Release management**

A system release is a version of a software system that is distributed to customers. For mass- market software, it is usually possible to identify two types of release namely major releases, which deliver significant new functionality, and minor releases, which repair bugs and fix customer problems that have been reported.

***SCM Tools***

SCM tools are software tools that automate and facilitate the application of the SCM best practices. As with a compiler, debugger, and editor, an SCM tool is an essential part of every software engineer’s tool kit today.

***SCM Processes***

A process defines the steps by which you perform a specific task or set of tasks. An SCM process is the way SCM is performed on your project—specifically, how an SCM tool is applied to accomplish a set of tasks.

***Summary***

In this question we define software configuration management in simple terms as the mechanisms used to control the evolution of a software project. An understanding of what we mean by software configuration management is crucial because if we don’t know what we want to do, we have no hope of converging on a good software-development environment. To enable this understanding, we specifically discussed software-development best practices and how they are enabled by a good SCM system. We also introduced the concepts of the SCM tools and processes that are used to implement those best practices.

1. ***Write the difference between version and release?***

***Version***

The version refers to the set of customization parameters. The MerlinCryption program used to decrypt an encrypted file must have the same version as the MerlinCryption program used to encrypt the file. If the versions are not the same, an error message Version mismatch is written to the log file (and displays if there is no prior -b switch), and decryption does not occur.

MerlinCryption offers the ability to create guaranteed unique custom versions of all its encryption/decryption products. All custom versions are guaranteed to be incompatible with the standard version and every other custom version.

To find out the version of your software, on the command line include the -v switch. The standard version will then output:

Ver.: 20110416az

***Release***

The release information is the release number and date of the software product. A change in release number and date does not alter version compatibility in any way. The release identifies features that may have been changed or added to the product since the previous release.

To find out the release of your software, on the command line include the -r switch on the command line. Listed below is a sample of the release output:

Release 6.0 - 20110921