**EMB0000330c8383Chapter 25 : Configuration Management**

**25.1**

**Problems that could arise if company does not have effective configuration management policies:**

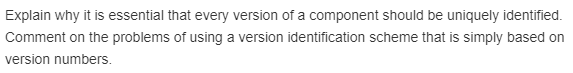
**1. New versions of software systems cannot be created effectively as they change. Developers cannot keep track of the changes to the software.**

**2. Controlling the costs and effort involved in making changes to a system is difficult.**

**3. Wrong version of a system may be delivered to the customers or forget where the software source code for a particular version of the system or component is stored**

**4. If someone leaves the company, protecting investments in software and the ability to reproduce a build with the correct components or continue development on a project is difficult.**

**5. Ineffective quality management process because configuration management may be seen as part of a more general quality management process.**

**25.2**

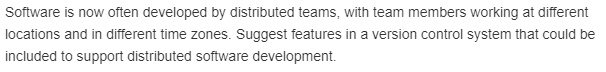
**In a large software system, there are hundreds of software components, each of which may exist in several different versions. There must therefore be an unambiguous way to identify each component version to ensure that the right components are included in the system. A consistent identification system is important because it simplifies the problem of defining configurations.**

**Version numbering is one of the most commonly used version identification scheme in which the component is given an explicit, unique version number. In this scheme a lot of extra information is to be maintained. When there are more number of versions, the numbering goes high and also sometimes confuses.**

**EMB0000330c838525.3**

**If different components are modified simultaneously at a same time, there may be several independent sequences i.e. many codelines for branches are developed. When these codeline branches are merged to create a new version of the component that includes all changes that have been made, there are overlaps between the changes made and they interfere with each other. The changes when merged may not be compatible with the existing system. If the changes made to the components are different, then the component versions may be merged automatically by version management system.**

**25.4**

****

**Distributed development of software**

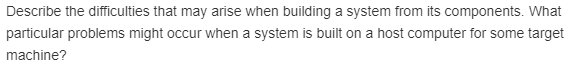
**The development of software involves analyst, developer, coders, testers, technical writers. It is possible that they all are working from different locations and time zones. So, the approach to develop any software which is developed across the globe to fulfil its requirements is known as distributed development of software.**

**When any updating is required in any software, then, its updated version is released, this process is known as version management. The points to show the features which must be included in version controlled system in distributed development environment are as given below:**

**• The unique identifier of software must be assigned. By doing so, the different versions of any software can be managed under single name.**

**• The version controlled system maintains the record of changes from previous version to new version. So, history of records must be updated.**

**• It is possible that different persons across the globe are working for release of new version controlled system. You must have to check that the newly made changes are not affecting the overall performance of system.**

**25.5**

**Problems with system building:**

**• In a system with a large number of components, there are chances for some components to be missed.**

**• When the specification of version is inappropriate, the system may fail to work after some duration.**

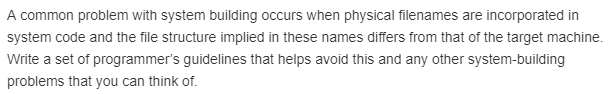
**• Naming in code may cause some problems because naming conventions may differ in each place.**

**• Different compiler versions may actually generate different code and the compiled component will exhibit different behavior.**

**When a system is built on a host computer for some target machine, it is not possible to build and test the system on the development computer, because the target environment may be either small and simpler than the development environment or the target environment is large enough which include databases and other COTS systems that cannot be installed on host computer.**

**EMB0000330c838b25.6**

**The software that was developed may not support the new version, and so it has to be maintained on the obsolete computers which support the software. Also it may be cost effective if it is maintained on the obsolete computers. Economic reasons may also require software to be maintained on an obsolete computer. It may be expensive to deploy the large software system on a new version computer. It can also arise when programs are developed in a programming language which is no longer used**

**25.7**

**The guidelines that help in avoiding the system building problems are:**

**1. Check if data files are referenced within a component have the same name that is used for the data file on the target machine?**

**2. Is the appropriate version of the compiler and other required tools available? Current versions of software tools may be incompatible with the older versions used to develop the system.**

**3. Have all the components that make up a system been included in the build instructions?**

**4. Has the appropriate version of each required component been included in the build instructions?**

**5. Are all required data files available?**

**EMB0000330c838f25.8**

**Change request form, will probably justify a change. Nevertheless, a – most likely simplistic –problem reports and system enhancements.**

**Key Benefits**

**• All change activity are arranged and controlled, thus resulting in fewer instances of loss of quality, normally associated with uncoordinated changes.**

**• Provides the ability to monitor and approve change requests from any single location.**

**• Provides the ability to customize your change request form to suit existing change management processes and terminology.**

**• Maintaining the integrity of the performance measurement baselines –all approved changes should be reflected in the Integrated Project Plan, but only project scope/requirements changes will affect the performance measurement baselines.**

**• Coordinating changes impact throughout the project plans.  For example, a proposed schedule change will often affect cost, risk, quality and staffing.**

**EMB0000330c839125.9**

**1. A form editor that allows change proposal forms to be created and completed by people making change requests.**

**2. A workflow system that allows the configuration management team to define who must process the change request form and the order of processing. This system should also automatically pass forms to the right people at the right time and inform the relevant team members of the progress of the change. E-mail is used to provide progress updates for those involved in the process.**

**3. A change database that is used to manage all change proposals and that may be linked to a version management system. Database query facilities allow the CM team to find specific change proposals.**

**4. A change-reporting system that generates management reports on the status of change requests that have been submitted.**

**5. Design and implement efficient procedures for the distribution and installation of changes to IT systems.**

**6. Plan your change implementations to include specific tasks that are appropriate for each change.**

**EMB0000330c839325.10**

**System release includes executable code, data files, configuration files, and documentation. And release management involves decision making on system release dates, distribution of information.**

**At the time of release, some of the factors are considered. Those are**

**1. Are all configuration files (which define how the release should be configured for particular installation) are included?**

**2. Have to create a new release of a software application when a new version of the operating system platform is released and check for right version of the system building tools are used or not.**

**3. Have all components been included?**

**4. Is the right version of all components been included?**

**5. Are there any problems with full path name references?**

**Apart from these some of the factors are influencing the system release planning.**

**Technical quality of the system:**

**System faults are reported that affect the way in which customers use the system and faults may be replaced by issuing patches.**

**Competition:**

**It is necessary for existing customers. Because, a competing product has introduced a new feature.**

**Marketing requirements:**

**The marketing department of an organization may have made a commitment for releases to be available at a particular data.**

**Customer change proposals:**

**Customers may have made and paid for a specific set of system change proposals, and they expect a system release as soon as these have been implemented.**

**Are all configuration files (which define how the release should be configured for particular installation) are included?**

**Have to create a new release of a software application when a new version of the operating system platform is released and check for right version of the system building tools are used or not.**

**Have all components been included?**

**Is the right version of all components been included?**

**Are there any problems with full path name references?**