Software Project Management
Prof. Durga Prasad Mohapatra
Department of Computer Science and Engineering
National Institute of Technology, Rourkela

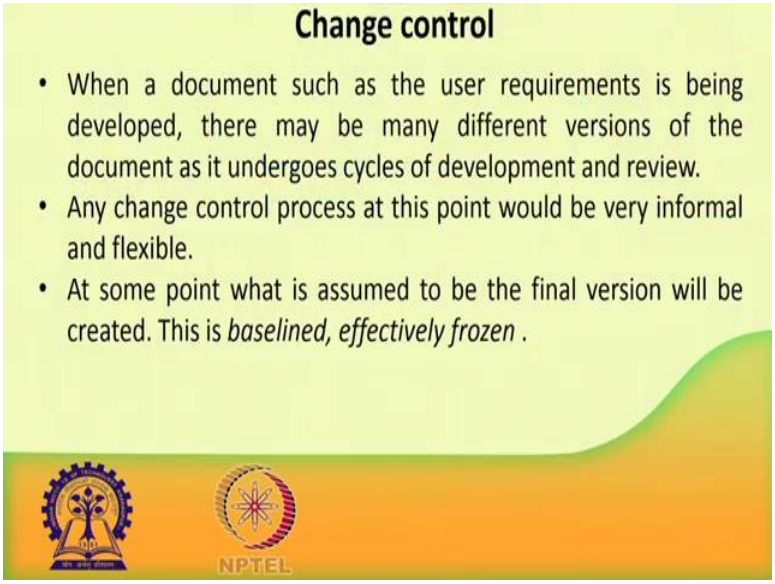
Lecture - 46
Project Monitoring and Control (Contd.)

Good afternoon. Now let us discuss another important aspect of Projective Monitoring and the Control that is known as change control and software configuration management.

(Refer Slide Time: 00:26)

First let us discuss about change control, then we will go to Configuration Management.

(Refer Slide Time: 00:28)



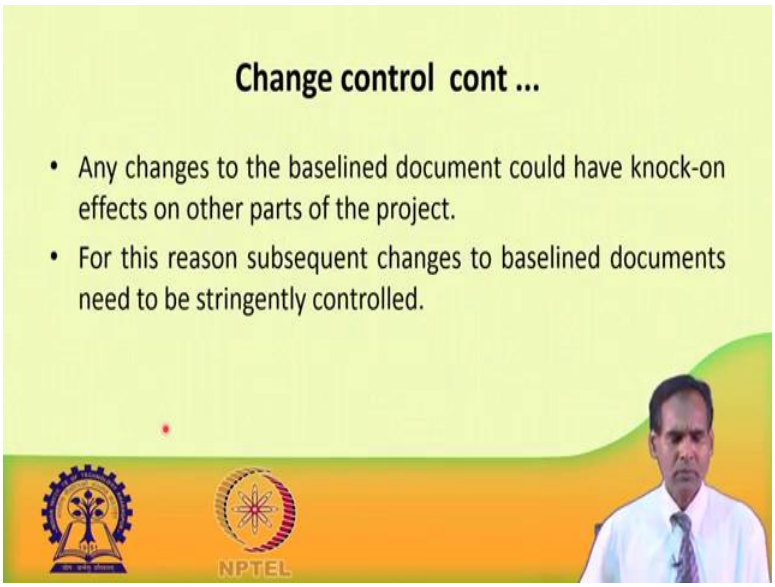
So, when documents such as a user requirement etcetera it is being developed, there may be many different versions of that document, because as it undergoes the cycles of or the different phases of software development process and review. Any change control process at this point would be very informal and it will be flexible and later on we will see the details.

At some point what is assumed to be final version that has to be frozen, at some point or some point will come when it is assumed that this document is final it is the final version or that has to be created; then we must what freeze that document development process there. So, this will be treated as the baseline version and this is all most effectively frozen at this point.

(Refer Slide Time: 01:19)

Baseline products are normally the foundation for the development of further products or further documents; for example, if you want to develop what the interface design documents, then you can develop it from the baseline user documents.

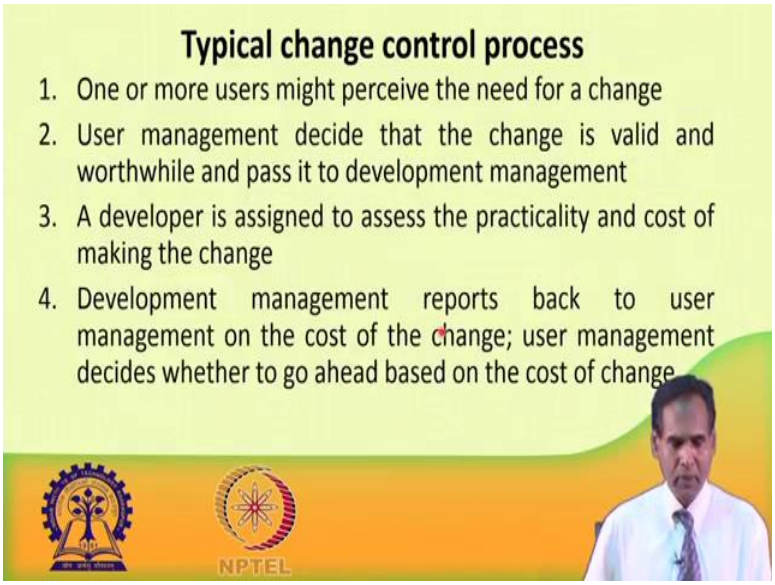
(Refer Slide Time: 01:41)



Now, any change to the baseline document it may create what a knock-on effects on other parts of the project; that means, the other parts of the project might be affected, if you will make any changes to the baseline document, but sometimes these changes are

required. So, that is why what for this reason, the subsequent changes to the baseline documents they have to be very stringently controlled and monitored.

(Refer Slide Time: 02:08)



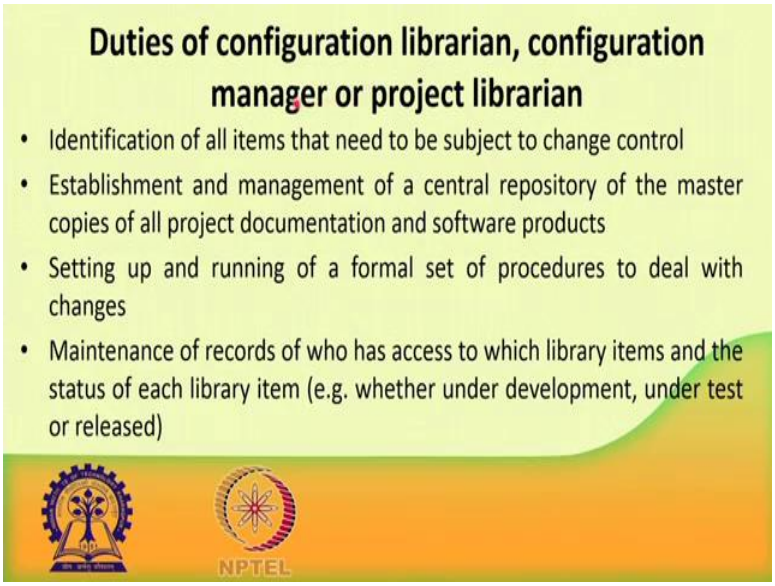
Now, let us see what are the typical change control process, what steps are followed in the change control process. First, one or more users they might perceive that there is a need for a change; then if they perceive that there is a need for the change. Then the user management, they have to decide that the change that is perceived by the users that is thought by the users it is valid and worthwhile and then they will pass it to the develop management. Next a developer from the development group, he will be assigned to assess what is the practicality and what is the cost of making the change; if it is practically feasible then they may go for to the next step.

Similarly, they will evaluate how much cost will be involved with the change. Then the develop management, they reports back to the user management that, yes this is the cost that will be required for the change and then the user management has to take a decision whether to go ahead with the change based on the cost. If the cost is high, they may not go further or if they cost which acceptable with them, they may give green signal yes you can go ahead for the change to the development team.

(Refer Slide Time: 03:17)

Now, then after the user management is agreed, then one or more developers they are authorized to make copies of the components they have to be modified. Then the copies are modified by these one or two developers; then after initial testing, a test version is released to the users for conducting acceptance testing by the users. Then the users they conduct the acceptance testing, when the user satisfied with the acceptance testing, yes the change is ok; then the operational release of that product or document is authorized, it is released and then the master configuration items they are updated with these new changed one's.

(Refer Slide Time: 04:01)



So, during this change control process, so in every organization normally one staff is there known as the configuration librarian. Let us see what can be the duties of the configuration librarian. So, first he should do the identification of all the items that need to be subject to change control; what items, what documents that need to be undergo a change control. He will also responsible for establishment and management of a central repository of master copies. So, all the master copies will be what stored in a central repository; so it is the job of the configuration librarian to for making arrangements for the establishment and management of a central repository of the master copies of all the project documents and software products.

Then he is also responsible for setting up and running up a formal set of procedures to deal with the changes; he should set up and run a formal set of procedures. Also, he is responsible for maintenance of the records of who has access to which library items and when and what is the status of each library item; like whether it is under development, or under test or it is released. So, those kind of things will be maintained by this configuration librarian.

(Refer Slide Time: 05:27)

Now, let us see the important aspect that is the software contribution management. So, normally changes in a project, it can take place in any of the work product such as requirement document, SRS document, design documents, coding test plans, etcetera and this may be due to many regions such as the bug fixing, changes on account of the work

simplification and considering the efficiency issues, etcetera; you might have to what change these various documents.

So, change management can be done actually manually, it can be done manually by a designed configuration librarian. In the last slide, I have told what are the duties of a configuration librarian, the configuration librarian can manually what do the change management. But the manual change management process gets overwhelmed it very much difficulty and time consuming; when we consider the changes taking place on all work products.

So, when all the work products you have to in a project there are several work products and if you want to change all the work products; then it is very much difficult to do it manually. Also when there are multiple variants of the product were there, multiple variant, several variants of the product are there and then several persons may simultaneously work on these multiple variant they may try to change, then this will create problem. So, that is why manual change management is very much difficult for this, you have to go for automating it.

(Refer Slide Time: 06:58)

So, in this situation, a systematic software configuration management process called as SCM, it with appropriate tool support needs to be deployed. SCM also the software configuration management is concerned with tracking and controlling changes to a

software. So, software configuration management, it is concerned with how to track and how to control the different changes made to a software.

In any systematic development and maintenance environment; various work products may require continually change such as a code, design document and SRS documents etcetera, associated with the software; they may require continual, they may continually change during the development phase as well as maintenance phase and you have to systematically manage these changes.

(Refer Slide Time: 07:52)

So, in a team development environment what will happen, each member of the development or maintenance team they would be decide they will be assigned to handle some modification requests. So, both from the development team as well as management team, maintenance team, so some members will be assigned to handle the modification request. Therefore, every work product would have to be accessed and modified by several members.

So, every work product they have to be accessed first, whether it is now possible to change it or not; if it is possible that you can make a change it is viable, then it can be what modified by several members. But if several members simultaneously they want to modify a single document, then problem will be there, it will be it may lead to inconsistency. In such a situation, unless you use a proper configuration management

system, then several problems can appear. We will see what problems can appear, if you will not use a proper configuration management system.

(Refer Slide Time: 08:54)

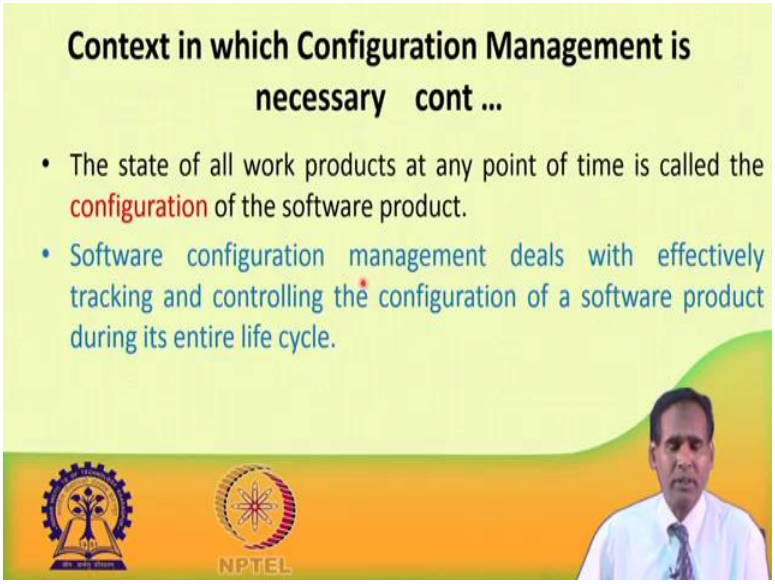
So, let us see what is configuration management, as I have already told you that configuration management deals with, it concerned with tracking and controlling the changes to a software. So, we can define configuration management like this, it is the set of activities through which the configuration items are managed. We will see what are configuration items in the next slide. So, configuration management is the set of activities through which the different configuration items they are managed as well as maintained as the product undergoes it is lifecycle phases.

(Refer Slide Time: 09:31)

Now, let us see what are the context in which configuration management is necessary. So, two important phases will use configuration management during the development phase as well as during the maintenance phase. During the development phase, the work products they may get modified as a development activities are carried out, so you have to what you go for configuration management.

Similarly during the maintenance phase, the work products may change due to various types of enhancements and adoptions that are carried out including the what bug fixing etcetera. So, for this also you may have to go for changing the, what you may have to go for changing the configuration. So, change also configuration management is also necessary at this point of time. So, thus the state of the work products they continually change both during the development period as well as the maintenance period.

(Refer Slide Time: 10:29)

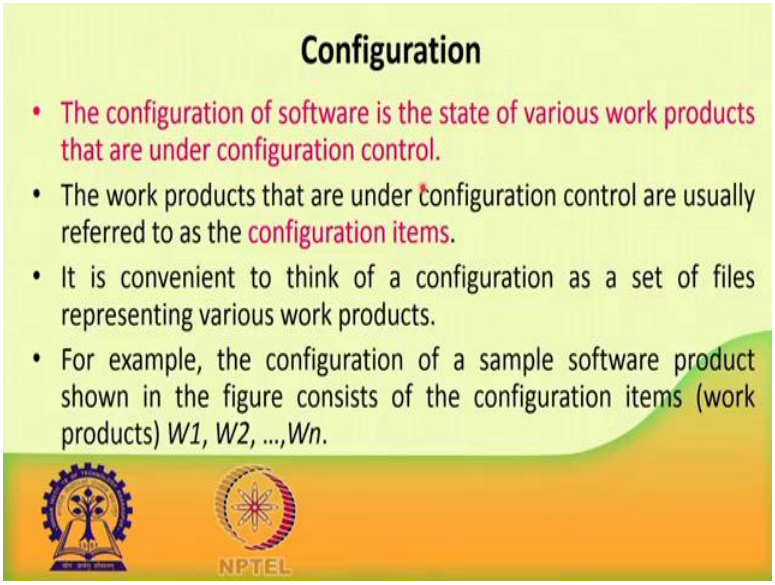


Now, let us see what do you mean by configuration? The state of all work products at any point of time is called the configuration ok. The state of all the work products such as SRS document and these who can say that design documents, coding, test plans, etcetera are the state of all work products at any point of time is called the configuration of the software product. Software configuration management deals with I have already told you that, software configuration management deals with effectively how to track and how to control the configuration of a software product during its entire lifecycle.

(Refer Slide Time: 11:06)

For effective configuration management, it is necessary to deploy a configuration management tool. There are many configuration management tools available out of them, some are open software, that is there you can get them free of any licensing fees and others are commercial tools; towards the end I will give a list of some commercial tools or some free open source tools for software configuration management. So, configuration management practices it includes two important things; version control and the establishment of the baselines.

(Refer Slide Time: 11:41)

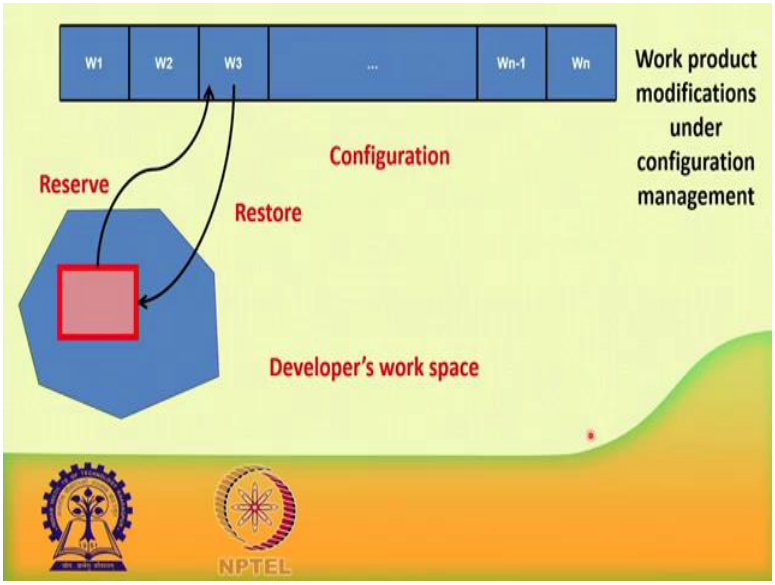


Now, let us see some of the important terminologies associated with configuration management. First configuration, I have already told you that, configuration of software means it is the state of the various work products those are under configuration control ok.

So, configuration means, configuration of software means it is the state of various work products such as SRS document, design documents, codes or codes test plans, etcetera that are under configuration control. The work products that are under configuration control are usually referred to as configuration items ok. So, these work products which are under configuration control they are termed as contribution items. It is very much convenient to think up a configuration as the following; you can consider them as a set of files representing various work products this may be thought of as configuration.

For example the configuration of a sample software product which we have shown in the next slide, it consisting of the following configuration items or the work products like this.

(Refer Slide Time: 12:47)



This is a says work product. So, W1 it is divided into so many configuration items, here the configuration items are W 1 W 2 W 3 up to W n etcetera.

(Refer Slide Time: 12:59)

Now, let us see the next term that version; as development and the maintenance activities are carried out on a software product, it is a configuration keeps changing. So, I already

told you during development period as well as maintenance period, so whenever they are carried out on a software product, it is configuration that is one or more of the configuration items they keep on changing. It is often necessary to refer to the configuration that existed at certain point of time; we want to know that, what is the configuration say before two days like that.

So, very often it is necessary to refer to the configuration that existed that the configuration of the software that existed at certain point of time. For example, say we want to refer to the last week's configuration of the software or last year's configuration of the software. So, in that way we might have to refer the configuration at some point of time. Therefore, a version means it is a configuration that existed at certain point of time.

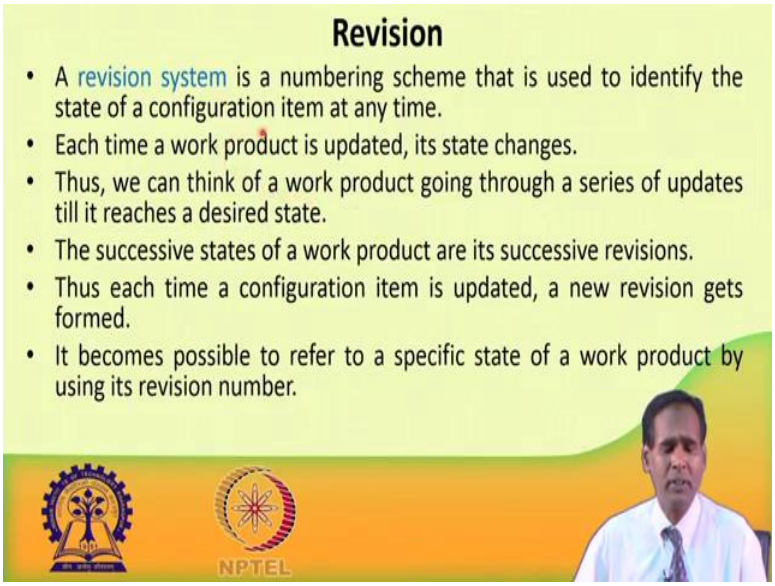
(Refer Slide Time: 13:53)

In very briefly we can say, that a version is a configuration that existed at some point of time at certain point of time. So, more technically versioning is a numbering scheme that help us, identify a specific configuration at a certain point of time, ok. So, technically we can say that versioning is means it is a numbering scheme. Why it is required? That helps us identify a specific configuration at a specific or at a certain point of time. How this can be achieved, how this numbering scheme can be achieved, how this versioning or numbering scheme it can be achieved? This can be achieved by a configuration tool by tagging the files representing or yes representing the configuration items with the version name.

(Refer Slide Time: 14:53)

As a software is released and used by the customers; many errors are discovered, and then enhancements are made to what improve, but the functionalities. So, a new release of the software, then several releases of the software come; a new release of the software is an improved system which is intended to replace an old one. So, you know that in many versions are coming, many releases of the software coming may be Microsoft; so a new release of the software it is an improved system which is intended to replace what the older one. So, usually a product is described as version m and release n or simply as a version m dot n; where m refers to the version and n refers to the release.

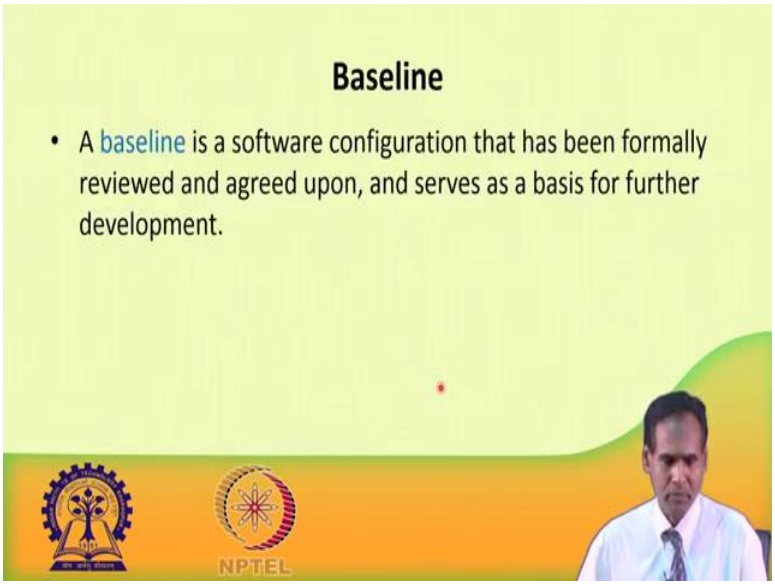
(Refer Slide Time: 15:45)



Now, let us quickly see about what is a revision. A revision system is a numbering scheme that is used to identify the state of a configuration item anytime. So, because you must be clear about the difference between what version and revision; a revision system is also a numbering scheme that is used to identify the state of a configuration item at any time. Each time a work product is updated, it is state; obviously, it changes. So thus, we can think of a work product which is going through it is series updates till it reaches a desired state. So, till reaching a desired state a work product may go through a series of updates. So, the successive of states of a work product are it is successive revisions.

So, whenever you are making any updation, updates, so the successive states of a work product they are it is a successive revisions. So, thus each time a configuration item is updated, any time any configuration items such as code or design document or a SRS document is updated a new revision gets formed. Then it becomes possible to refer to a specific state of a work product by using it is revision number. Every time you are when you are making upgradation, a number you are assigning it becomes possible to refer to the specific state of a work product by using it is what revision number.

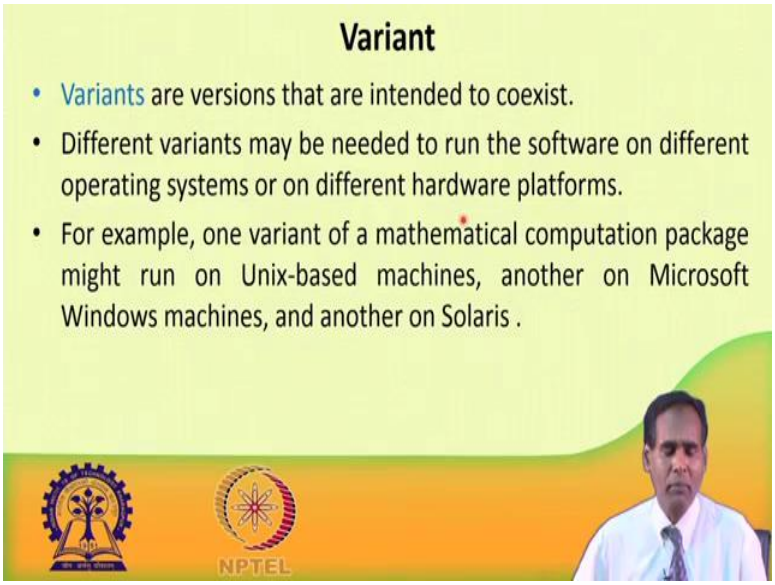
(Refer Slide Time: 17:09)



Now let us see what is a baseline? Baseline is a software configuration that has been formally reviewed and agreed upon and it serves as a basis for further development. So, here the what, first this a software configuration that has already been formally reviewed by the developer also the users have seen in it and they have agreed on, both of them

have agreed upon it and then it can serve as a basis for further development of the software products.

(Refer Slide Time: 17:41)



Variants; variants means they are the versions that are intended to coexist. So, these are the different versions they are intended to coexist will. And different variants may be needed to run the software on different operating system ok. It might be required that different variants, they might be required to run the software on different operating systems or on different hardware platforms, etcetera or maybe on different browsers.

So, let us take a small example that one variant of a mathematical computation package might run on Unix-based machines, another may run on Microsoft windows machines, another variant may run on Solaris etcetera. So, this could be one example. Another example could be those who have used LaTeX, so LaTeX normally it is used in where in Unix version; but another variant is they are, if you are using windows version the other variants is called a MiKTeX. So, that is a text processing software. So, you can use in Unix variant what LaTeX and for windows the variant is with MiKTeX.

(Refer Slide Time: 18:47)

So, variants may also be required to be created when the software is intended to be used with different levels of sophistication of the functionalities; for example, for novice users it is the novice version, for enterprises it is enterprise version and for experienced versions you can say or for professionals the version may be professional version.

So, variants are often created during the operation phase, during the development phase, and as and when the software products with overlapping functionalities they are required, and then you can create different variants. Even the initial delivery of this software might consist of several versions. So, whenever you are initially, first time you are delivering, the initial delivery of software might consider up several versions and more variants may be created later on.

(Refer Slide Time: 19:32)

Now, let us see what are the purpose of SCM why what software configuration management is used, what is the purpose. So, there are several purposes, some important purposes I have written here that why software configuration management is required; first, problems associated with concurrent access.

So, one configuration item, simultaneously many users or many developers they are trying to access then, it will lead to inconsistency. So, in order to avoid the problems associated with concurrent access, we should go for software configuration management. Similarly undoing the changes, several times what is required, we may have to we have done something wrong thing and we want to undo it. So, in order to undo the changes, this, what software a configuration management is required. Another is system accounting, in order to keep track of that who had made the change, when he had made the change and why he had made the changes.

So, to keep track of all the details, that is you that we call in technology the system accounting. So, that can be another reason why SCM is required. Similarly handling the different variants and helping fix bugs in them, I have already told about variants, how to handle the variants, how to tackle the variants and how to what help the, how to what fix the bugs in the variants, for them also you can use this software configuration management.

Similarly, accumulated determination of the project status, what is the current project status, we want to know the accurate what status of the project. So, for that, you can also use the software configuration management. And similarly preventing unauthorized access to the work products, see only the project manager he will give the permission to the developers to access the work products.

Without the permission anybody or anybody else cannot actually access this. So, if anybody is trying to access the work products unauthorized way, then that will be prevented; if you are having software configuration management. So, this is another reason that preventing unauthorized access to the work product this can be achieved by using software configuration management. So, these are the different reasons purposes of using software configuration management, these things have been explicitly what explained here, I am skipping this thing, these are easy things you can read there yourself.

(Refer Slide Time: 21:56)

Now, we should see that what is the, what we have seen so far, we have first seen that, what is the need of change control, why change control is at all required. Then we have also seen that the steps of the change control, how the change control can be achieved in an organization. We have also discussed the change control procedure; we have also explained the concept of software project management, what is software sorry what is software configuration management, what is the need, what are the purposes. We have

also discussed what the context in which it is necessary, in which context, what the software configuration management is necessary.

For example I have told that it is required at the development phase also, it is required at the what maintenance phase, also what are the different purposes just I have already told you that, the important purposes for which software configuration management we required. We have also defined various terms, the common terms which as a what as a student you must know that what are the different terminologies used in configuration management such as what is the configuration, what is a configuration item, what is a baseline item and what is revision, what is a release.

So, those kind of basic things also we have discussed here. So, actually next actually we must see about, we have seen now the basic concept of configuration management; then we must see about how to do it, how to what process, you should follow what, for doing this configuration management. I have already told you doing configuration management manually by a configuration librarian is very very difficult, you must require you have to automate it.

So, what are the tools there available for automating this. So, I will tell that some of the tools are there for doing this and some tools are freely available, some tools are what commercial tools; for example, freely available tools will use SCCS and RCS. So, those are details of the tools we will discuss in the next class; but first we will discuss about what are the steps they must be followed for achieving this or for making software configuration management and those steps will discuss in the next class.

(Refer Slide Time: 24:36)

And these things we have mainly taken from these two books ok.

Thank you very much.