UNIT V

CHAPTER 10

Project Risk Management

Syllabus

Project Risk Management : Risk Analysis & Management: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Risks Monitoring and Management, The RMMM plan for case study project

Syllabus Topic : Risk Analysis and Management

10.1 Risk Analysis and Management

SPPU – May 14, May 15, Dec. 15, May 16

University Questions

Q. What are the types of risks ? Explain in brief. **(May 2014, 6 Marks)**

Q. Explain principles of risk management in detail. **(May 2015, 8 Marks)**

Q. What are the different categories of risk. Explain risk management process in detail. **(Dec. 2015, May 2016, 8 Marks)**

- Whenever we start any business or any development process, we take into consideration the risks involved in accomplishing that task. Similarly when software development process is started, it is main job of a software manager to look into all types of possible risks involved in the entire process.
- It involves focusing on the possible risks that could affect the project development process :
  - Schedule of the development process
  - Quality of the application under construction
- It is the responsibility of a project manager to look into the matter and take necessary action to avoid these risks. All the results of risk analysis and analysis of consequences of risk occurring should be well documented in the planning of the project itself.
- If the risk management is effective, then it becomes easier to handle all the problems and it is ensured that the project schedules and budget are within acceptable limits and there is no schedule slippage and ultimately no budget slippage.

10.1.1 Software Risks

- Software risk is a type of risk that always threatens the project development processes and the software under construction.
- Following are three important categories of risk :

1. Project risks
2. Product risks
3. Business risks

1. **Project risks**
   - These are the risks that directly affect the schedule of the project and the resources involved in the development process.
   - **Example of project loss :** Loss of an experienced developer and designer.
2. **Product risks**
   - The product risks affect the quality and performance of the application built.
   - **Example of product risks :** Failure of a purchased component to perform as per expectation.
3. **Business risks**
   - The business risks are affecting the organization those develop and process the software.
   - **Example of business risks :** A competitor of the organization introducing a new product.

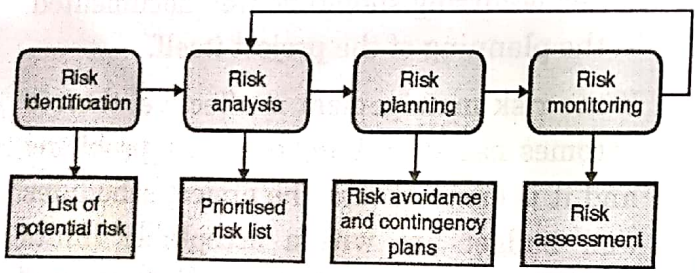

Fig. 10.1.1 : The Risks management process.

10.1.2 Reactive Versus Proactive Risk Strategies

1. **Reactive risk strategy**
   - In this type of risk strategy, the project is monitored closely for the likely risks that may occur.
   - Resources are monitored to guess the possible risks and deal with them so that they do not become the problems for budget slippage and cost slippage.
   - Generally in reactive risk strategy, the development team members are directly not involved in the risks occurring.
   - And after occurrence of the risks, the development team members do some rapid action to overcome the problems.
   - This rapid action is called as **"fire fighting mode"**.
   - After failure of all these actions, the concept of "crisis management" comes into picture and the project is in deep trouble and uncertainty happens.
2. **Proactive risk strategy**
   - A proactive risk strategy is initiated when the actual technical work begins.
   - The potential risks are identified and the probability and impact are assessed so that the development team members establish an appropriate plan to manage all these risk.
   - The main goal to avoid the risks initially.
   - But all the risks cannot be avoided, therefore the development team creates a contingency plan that will control the risks in an effective manner.

Syllabus Topic : Risk Identification

10.2 Risk Identification

SPPU – Dec. 12, May 13, Dec. 13, Dec. 16

University Questions

Q. Write short note for following : Risk identification. **(Dec. 2012, 4 Marks)**

Q. Explain the risk identification. **(May 2013, 8 Marks)**

Q. Explain the risk identification and assessment process for a software project ?

Q. What is risk identification ? What are the different categories of risks ?
(Dec. 2016, 7 Marks)

- Risk identification is related to discovering possible risks to the project. It is a systematic attempt to specify threats to project plan.
- There are two types of risks :

1. Generic risk
2. Product specific risk

1. Generic risk

They are potential threat to every software project.

2. Product specific risk

- The project plan and software statement of scope are examined to find out threats due to special characteristics.
- The check list can be created of risk and then focus is subset of known and predictable risks in following subcategories :

(i) Product size
(ii) Business impact
(iii) Customer character
(iv) Process definition
(v) Development environment
(vi) Technology to the limit
(vii) Staff size and experience

(i) Product size

Risk involved in the overall size of the software.

(ii) Business impact

Risk involved in constraints imposed by management or the market.

(iii) Customer characteristics

Risk involved in the sophistication of the customer and ability of the developer to communicate the customer properly.

(iv) Process definition

Risk involved in defining the software process followed by software development organization.

(v) Development environment

Risk involved in availability and quality of the tools used in the development process.

(vi) Technology to the limit

Risk involved in complexity of system and risks associated with the new technology used.

(vii) Staff size and experience

Risk involved in overall development teams i.e. experience of developer, skill set of team members.

10.2.1 Assessing Overall Project Risk

- Following are the set of questions obtained from some previous successfully completed projects :
  - Are the top software managers and customer manages are committed to help the project ?
  - Are the end-users are involved in the development process with enthusiasm ?
  - Are all the requirements elicited properly by the customers and are these questions well understood by the development team ?
  - Have the customers involved completely in the development process with development team while defining the requirements ?
  - Are all the end users mentioning their realistic expectations from the project ?
  - Is the project scope stable or not ?

- Are there in the development team proper blend of skilled team members ?
- Are the requirements of the project stable ?
- Are the project team members well familiar with the latest technologies ?
- Are there enough team members in a team allotted for a project ?
- Are all the customers and users agree on the requirements of the project to be built ?

- If these questions are answered negatively, then the project may be at risk. In another words, we can say that the degree of risk is directly proportional the number of negative answers given.

10.2.2 Risk Components and Drivers

- In the project development process, it is essential for a project manager to identify the factors that affect the following **risk components** :
  - Performance
  - Cost
  - Support and
  - Schedule
- The factors that affect the risk components are also called as **risk drivers**. In the context of risk management, the above components can be explained as follows :
  - **Performance risk :** It is defined as the degree of uncertainty that an application satisfy the requirement expected by the customer.
  - **Cost risk :** It is defined as the degree of uncertainty that a project does not exceed its budget.
  - **Support risk :** It is defined as the degree of uncertainty that the final product will be adaptable and easy to maintain.
  - **Schedule risk :** It is defined as the degree of uncertainty that the final product will be delivered on its deadline as mentioned in the documents.
- The impact of the risk drivers on risk components are classified in four different classes as follows :
  - Negligible impact
  - Marginal impact
  - Critical impact, and
  - Catastrophic impact
- We can illustrate all these risk components (i.e. performance, cost, support and schedule) in the following chart :

Table 10.2.1 : The table showing impact analysis

Risk component / Class of impact	Performance	Cost	Support	Schedule
Negligible impact	If performance failure occurs, then it will cause inconvenience.	Errors can cause no serious impact on the project budget	The technical performance is not compromised. It can be supported with ease.	The delivery is before the deadline.

Risk component / Class of impact	Performance	Cost	Support	Schedule
Marginal impact	There is some degradation in technical performance.	Some schedule slips can cause the cost within limits.	Some small degradation in performance.	The delivery is within the limits.
Critical impact	There is question mark in completion and success of the project.	The failure can cause project delays and thus financial overruns.	Some delay in project delivery due to modifications in the LOC.	There is schedule slippage due to LOC delays.
Catastrophic impact (Disastrous impact)	If performance failure occurs, then simply it is the project failure.	Failure can cause drastic cost overrun and project delays.	The final product is non-responsive and can not be supported.	Unachievable deadline due to increased cost and financial crunches.

Syllabus Topic : Risk Projection

10.3 Risk Projection

- Risk projection is interchangeably called as Risk estimation also. The risk projection rates the risk in following two ways :
  - The probability of risk occurrence and
  - The consequences of the risk occurred.
- Following are four important risk projection activities that every manager, developer should perform :
  - Make a scale to measure the likelihood of the risk.
  - Describe the consequences of the risks.
  - Estimate its impact and
  - Write down overall accuracy of the risk projection to avoid misunderstandings.

10.3.1 Developing a Risk Table

SPPU - Dec. 13

University Question

Q. Write short notes for the following : Risk table. **(Dec. 2013, 4 Marks)**

- A risk table gives very useful technique to project managers for the risk projection. Following Table 10.3.1 is an example of risk table :

Table 10.3.1 A sample risk table

Risks	Category	Probability	Impact	RMMM
Size estimate may be significantly low	PS	60%	2	
Larger number of users than planned	PS	30%	3	
Less reuse then planned	PS	70%	2	
End-users resist system	BU	40%	3	
Delivery deadline will be tightened	BU	50%	2	
Funding will be lost	CU	40%	1	
Customer will change requirement	PS	80%	2	
Technology will not meet expectations	TE	30%	1	
Lack of training on tools	DE	80%	3	
Staff inexperienced	ST	30%	2	
Staff turnover will be high	ST	60%	2	

- In the Table 10.3.1 the impact values are as follows :
  - For the disastrous situations → 1
  - For some critical situations → 2
  - For the average situations → 3
  - For the negligible situations → 4
- In the above table the categories used are as follows :
  - PS used for project size risk
  - BU used for business risk
  - CU used loss of funds etc.
- The risk table is prepared right in the beginning of the development process. It is very much useful in analyzing the various types of risks associated. The risks are categorized in different categories as shown in Table 10.3.1.
- The probability of occurrence of particular risks is also estimated and associated impact values are listed in the above table. The project manager goes through this table and gives a cutoff line. All the risks that fall below the cutoff line are evaluated again. Similarly the risks that are above the cutoff line are properly managed by the project manager.

10.3.2 Assessing Risk

Following are three important factors that affect the result if risk occurs :

- **The nature of risk :** It is important to know the nature of the risk expected to occur. For example, if the hardware interface is not defined properly, then it will cause the problems during integration.
- **The scope of risk :** It looks into the severity of the risks and how the end-users will be affected by the occurrence of the risk.
- **The timing of risk :** It is important to note that for how long the impacts of risk will remain.
- Consider the following risk analysis approach. In order to determine the impact of risk, following steps are recommended :
  - Find the average probability of occurrence of the risk for the risk components like performance, cost, support and schedule risk.
  - Refer Table 10.2.1, and calculate the impact of each component.
  - Complete the risk table and perform analysis on the result.
- Let RE is the overall risk exposure and it written as :

  RE = P x C

Where P→probability of occurrence of a risk

C→the cost incurred to the project, if risk occurs.

Consider the following example to understand above equation :

Example 1 : Let a software team describes the probable risk in the following manner :

(a) **Risk identification :** There are 75 % reusable components that can be integrated to the complete application. The remaining 25 % are the functionalities that are customized.

(b) **Risk probability :** Let there are 85 % probable risks.

(c) **Risk impact :** Let there are only 65 % reusable components planned out of 75 % used. Then in this scenario, nearly 15 components have to be developed from scratch, i.e. they are not reusable components. For calculation purpose, we consider 100 LOC on an average. Assume that $20 are incurred for each of the LOC, the total cost to develop the complete project can be computed as :

$15 \times 100 \times 20 = \$30,000.$

Thus the **risk exposure** can be computed as follows :

$$RE = 0.85 \times 30,000$$
$$= 25,500 \text{ (Approx.)}$$

In this way, the RE can be calculated for each of risk in the risk table.

Benefits using risk exposure

- The risk exposure is useful in adjusting the final cost of the project.
- It is also useful in cost estimation of the entire project.
- It is also useful in some situation where there is risk of schedule slippage. It can predict the probable risks.
- If the risk is predicted, then easily the organization can increase the manpower in that project to recover the schedule slippage.

10.3.3 Project Plan

- Planned and controlled software projects are conducted for one and only reason because it is the only way known to manage complexity.
- Project failure rate remains higher than it should be even though the success rate for software projects has improved. One of the reasons is that the customers lose interest quickly (because what they have requested was not really as important as they first thought), and the projects are cancelled.
- The software engineers and the software project manager must follow certain guidelines for project plans in order to avoid project failure :
  - Carefully design a set of common warning signs,
  - Understand the critical success factors that lead to good project management, and
  - Develop a common sense approach for planning, monitoring, and controlling the project.

Syllabus Topic : Risk Refinement

10.4 Risk Refinement

- In the beginning of project planning, a risk may be quoted in general but with the progress of time, one can be able to understand the depth of the risk. So it is possible to refine the risk into a set of more detailed risks, and it would be more convenient to mitigate, monitor, and manage.
- CTC (condition-transition-consequence) format may be a good representation for the detailed risks. The example of CTC format is given below :
- Given that <condition> then there is concern that (possibly) <consequence>.

- This general condition can be refined in the following manner:
- **Sub condition 1 :** Certain reusable components were developed by a third party with no knowledge of internal design standards.
- **Sub condition 2 :** The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components.
- **Sub condition 3 :** Certain reusable components have been implemented in a language that is not supported on the target environment.
- Even though we refine the risk into more detailed risks, but the consequences for the risks remains same. It only helps in separating the part of the risk for better understanding and analysis.

Syllabus Topic : Risk Mitigation, Risk Monitoring and Risk Management (RMMM)

10.5 Risk Mitigation, Risk Monitoring and Risk Management (RMMM)

SPPU - May 12, May 14

University Questions

Q. What is Risk Mitigation Monitoring and Management (RMMM) ? Write a note on it. **(May 2012, 10 Marks)**

Q. Write short note on : RMMM. **(May 2014, 5 Marks)**

- Risk analysis actually helps the project development team to build a strategy to handle all possible risks. Following are three important issues (or steps) that must be considered for developing effective strategies :
  - Risk avoidance (i.e. Risk mitigation)
  - Risk monitoring
  - Risk management and planning.
- In order to avoid the risk, the best approach is proactive approach. In the proactive approach, the development process starts with risk mitigation plan and it helps in avoiding possible risks.
- The step is risk monitoring that begins from the start of the development process. The project manager is generally responsible for keeping vigilance on the factors that can cause risk to occur.
- The project manager starts monitoring with the information received from the risk mitigation step. He scans all the documents prepared in the risk mitigation step.
- The third and final step is risk management and planning which assumes that the risk has occurred and accordingly the manager has to take necessary action.
- The Risk Mitigation, Monitoring and Management (RMMM) steps incur extra project cost.

10.5.1 The RMMM Plan

- Risk management strategies should be included in the project plan itself. The risk management plan is usually added as a separate plan in the project plan. This is referred as risk mitigation, monitoring and management plan or RMMM plan.
- The RMMM plan is executed as a separate plan to evaluate the risk. Each of the risks are documented separately.
- The RMMM plan is well documented in the beginning of the project itself. Once the project begins, the mitigation and monitoring activities are started.
- The risk monitoring has main three objectives :
  - Check whether the predicted risks actually occur.

- All the risk assessment steps are properly followed, and
- Collect all the necessary information that can be useful for future risk analysis.

Following are the list of probable risks that may occur during project development process. Here in this section, we explain each of the risks mentioned below discuss them keeping RMMM Plan in mind :

- Computer Crash
- Late Delivery
- Technology will not Meet Expectations
- End Users Resist System
- Changes in Requirements
- Lack of Development Experience
- Lack of Database Stability
- Poor Quality Documentation
- Deviation from Software Engineering Standards
- Poor Comments in Code

(1) Risk : Computer Crash

(a) Mitigation

- The computer crash can cause the loss of important data and ultimately it will result in failure of the project.
- It is always recommended that software development organization should keep multiple copies of the data at multiple locations to avoid loss.

(b) Monitoring

The project manager must keep a watch on the working infrastructure and working environment before the actual development work starts.

(c) Management

Any inconvenience in working environment must be noticed immediately and a proper action should be taken to make the system stable.

(2) Risk : Late Delivery

(a) Mitigation

- The late delivery will result in approval from the customer. If the customer is reluctant to give the approval, then the development organization will get a bad image and it will affect the organization for getting future projects.
- So the important steps and precautionary measure are taken to timely delivery of the project.

(b) Monitoring

Continuous monitoring is required during the entire development process. The development schedule must be observed strictly and if any slippage occurs, it be noted seriously and necessary action should be taken for the recovery.

(c) Management

The delayed project can move to critical state and ultimately can cause project failure. The project manager must look into the matter seriously and should negotiate with the customer for extending the deadline as a final solution.

(3) Risk : Technology Does Not Meet Specifications

(a) Mitigation

The formal and informal meeting must be conducted with the customer to avoid this risk. It guarantees that realistic products are being developed as per customer's need.

(b) Monitoring

The proper and frequent communication is needed to understand each other and ultimately the requirement.

(c) Management

- If the idea of the project specification does not meet the requirement narrated by the customer, then the management team must take quick action to resolve this issue.

- A meeting must be conducted to understand the problems.

(4) Risk : End Users Resist System

(a) Mitigation

The application must be developed keeping the end user in mind. The user interface must be user friendly and convenient to operate.

(b) Monitoring

Any mismatch must be monitored immediately and it should be brought to development team.

(c) Management

- If the system is resisted by the user then the software must be evaluated completely and it is necessary to uncover the reasons and user interface should also be investigated properly.
- After uncovering the reasons for customer resistance, the immediate actions are required.

(5) Risk : Changes in Requirements

(a) Mitigation

To avoid the changes in requirement by the customer, it always better to keep the record of requirements given by the customer. There must be formal and informal meetings conducted between the development team and the customer.

(b) Monitoring

The meetings between the customer and the development organization must be conducted to understand each other and the requirement also.

(c) Management

To rectify any notified problem, a meeting should be conducted to discuss at the issue and the resolution of the issue.

(6) Risk : Lack of Development Experience

(a) Mitigation

The development team members must be updated and they should be well experienced to use the development tools needed.

(b) Monitoring

The team members must look into other members also to find any weak link in the chain. If such cases are observed, then it should be brought to management team to handle.

(c) Management

The experienced members must help those who are proving a weak link.

(7) Risk : Database is not Stable

(a) Mitigation

The development team should communicate the database administrator to keep it stable. All the functions, procedures and database operation must be operated error free.

(b) Monitoring

Each of the team members must monitor continuously the functioning of all database operations. If any inconvenience is reported, it must be brought o attention.

(c) Management

Any noticed problem must be resolved instantly.

(8) Risk : Poor Quality Documentation

(a) Mitigation

All the stakeholders of the project development must conduct meeting to formulate a good documentation. Any suggestions must be incorporated to enhance the quality of the documentation.

(b) Monitoring

To keep the testing and development in normal condition, the documentation must be referred time-to-time.

(c) Management

Any good opinion must be included in the documentation and any unnecessary topic should be removed.

(9) Risk : Deviation from Software Engineering Standards

(a) Mitigation

To avoid deviation from the standards, the development team must be familiar with the entire software engineering standards. They should have proper knowledge and understanding of the process.

(b) Monitoring

The technical reviews must be taken to determine any deviation.

(c) Management

If any deviation noticed, it should be addressed immediately and the management team must guide the development team to bring them on the right track.

(10) Risk : Poor Comments in Code

(a) Mitigation

A writing standard must be followed while coding. All the functions and sub-functions must be commented properly for better understanding of the code.

(b) Monitoring

The codes must be reviewed on a regular basis to determine any poor comments.

(c) Management

If any poor comments are observed, action must be taken to minimize the poor commenting and refining comments as necessary.

Syllabus Topic : The RMMM Plan for Case Study Project

10.6 The RMMM Plan for Case Study Project

- We consider the Case Study "Waste Management Inspection Tracking System (WMITS)" and present the RMMM plan in the following section.

10.6.1 The general overview of RMMM Plan for WMITS

- Our goal is to assist the project team in developing a strategy to deal with any risk. For this we will take a look at the possible risks, how to monitor them and how to manage the risk.
- For software development to avoid any risk both the developer and client have to work together. Client has to spend time with the developer in the beginning phase of the software development. If client decides to change the software, meaning if client wants to add some more functions into the software or to change the requirement, this will have major effect on the development of the software.

10.6.2 The Description of Risk for WMITS

Business Impact Risk

- This is the risk where concern is that of the not being able to come up or produce the product that has impact on the client's business.
- If the software produced does not achieve its goals or if it fails to help the business of clients improve in special ways, the software development basically fails.

Customer Risks

- This is the risk where concern is client's motivation or willingness in helping the software development team.

- If the client fails to attend meeting regularly and fails to describe the real need of the business the produces software will not be one that helps the business.

Development Risks

- If client fails to provide all the necessary equipment for the development and execution of the software this will cause the software to become a failure. So in other words customer has to be able to provide time and resources for the software development team.
- If all the requested resources are not provided to the software development team odds for the software development to fail rises greatly.

Employee Risk

- This risk is totally dependent on the ability, experience and willingness of the software development team members to create the working product. If the team members are not experience enough to use the application necessary to develop the software it will keep pushing the development dates until it's too late to save the project.
- If one or more members of the software development team are not putting in all the effort required to finish the project it will cause the project to fail. Employee risk is one of the major risks to consider while designing the software.

Process Risks

- Process risk involves risks regarding product quality. If the product developed does not meet the standards set by the customer or the development team it is a failure.
- This can happen because of the customer's failure to describe the true business need or the failure of the software development team to understand the project and then to use proper equipment and employees to finish the project.

Product Size

- This risk involves misjudgment on behalf of the customer and also the software development team. If the customer fails to provide the proper size of the product that is to be developed it will cause major problems for the completion of the project.
- If software development team misjudges the size and scope of the project, team may be too small or large for the project thus spending too much money on project or not finishing project at all because of shortage of finances.

Technology Risk

- Technology risk involves of using technology that already is or is soon to be obsolete in development of the software. Such software will only be functional for short period of time thus taking away resources from the customer.
- Since the technology changes rapidly these days it is important to pay importance to this risk. If customer request use of software that soon to be obsolete software development team must argue the call and have to pursue customer to keep-up with current technology.

10.6.3 Risk Mitigation, Monitoring and Management for WMITS

Risk Mitigation

- It is important to have **mitigation plan** to avoid risks once and for all. Goal is to attack the risk even before it comes into existence.
- The plan will help in identifying the possible risks and to monitor them.
- In this risk concern is of under or overestimation (mainly underestimation) of the number of Function Points (FP). If we

estimate too few LOC (line of code) necessary for the project we may get wrong cost figures which can prove fatal to the software development plan.

- To avoid this from happening we will use conservative figures to reduce the probability of the risk. This means we will overestimate the LOC a little.

Risk Monitoring

- To monitor the risk here, we will keep track of the amount of functions necessary for the program throughout the entire development cycle. This will tell us if the project may come across risk in future.
- As **monitoring step** in this risk we will setup user meetings to show them the work that has been completed and to get user input on the work.
- We will have meetings every other week to present the work that has been done from the time of the last meeting. This will help team in staying in touch with the customers and will also be very efficient way to derive customer's input on the progress made.

Risk Management

- If a mistake has been made, user input on the completed work will provide us with information to fix or improve the software. We have done many meeting with the clients and plan to do meeting every two weeks; this should clear any misunderstandings between the software development team and customers.
- This is the best way to go at since the work that is done on the project is revealed during the meetings and customer gets chance to make adjustments necessary.

❑❑❑

UNIT V

CHAPTER 11

Software Configuration Management (SCM)

Syllabus

Software Configuration Management: The SCM repository, SCM process, Configuration management for WebApps, Case study: CVS and Subversion Tools, Visual Source Safe from Microsoft and Clear Case.

Syllabus Topic : Software Configuration Management (SCM)

11.1 Software Configuration Management (SCM)

SPPU - May 12, May 16

University Questions

Q. Define : SCM. **(May 2012, 4 Marks)**

Q. What is software configuration management (SCM) ? **(May 2016, 4 Marks)**

Review Question

Q. What do you mean by software configuration ? What is meant by software configuration management ?

- Basically the output of any software process contains the information based on various inputs. These output can be broadly divided into three important categories as follow :
  - The computer programs
  - The work products and
  - The data that consist of the information produced.
- All these information parts collectively called as software configuration.
- If each configuration item simply led to other items, little confusion would result. During the process, change takes place which may be unfortunate. But change can occur any time and for any undefined reason. The change is the only constant.
- The first law of system engineering says that "No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle".
- Following are four basic sources of change:
  - Every new business conditions and market conditions may cause the change in product requirement and business rules definitions.
  - Customers may require change in data presented by the information system. The customers may also demand various functionality delivered by their product of interest. The customers seek some new services. All these changes are customer oriented.
  - As the business grows or its downsizing may cause various changes in priorities of the project and also restructuring of the team happens.

- The budget and time scheduling constraints are also important factors for change in system or product.

- SCM is a set of activities used for managing the change during the life cycle of computer software. The software configuration management can also be considered as a software quality assurance activity during the development process.

11.1.1 SCM Basics (Configuration Management System Elements)

Review Question

Q. What are configuration management system elements ?

Four basic elements that should exist when a configuration management system is developed :

1. **Component elements :** The tools in the file management system uses the software configuration item.
2. **Process elements :** The process elements or the procedures uses effective approach towards the change management in engineering and use of computer software.
3. **Construction elements :** The automated tools are used in construction or the development process and ensuring the validated components should be assembled.
4. **Human elements :** In order to make the effective use of SCM, the team makes the use of various tools and process feature.

These elements are not mutually exclusive. For example, component elements work in conjunction with construction elements as the software process evolves. Process elements guide many human activities that are related to SCM and might therefore be considered human elements as well.

11.1.2 Baselines

- The change is the only constant in software development life cycle. The customer want to modify the requirement as the model gets ready. Since in the beginning, even customer is not fully aware of the product requirement. As the development begins, customers need lot of changes in the requirement.
- Once customers modify their requirement, it is now manager who modifies the project strategy.
- Actually as time passes, all the people involved in the product development process come to know exact need, the approach and how it will be done in the prescribed amount of time.
- The additional knowledge is required to know the exact requirement. It is very difficult for most of the software engineers to accept this statement that most of the changes are justified.
- The baseline is SCM that help in development process without affecting much the schedule and control the changes.
- The IEEE provides a baseline as follows :
- "A specification and requirement that is agreed upon between customer and developer is a basis for the product development and these requirements can be changed only through change control procedures".

11.1.3 Software Configuration Items

Review Question

Q. What are SCI's ?

- Basically SCI i.e. software configuration item is the integral part of software engineering development process. It is a part of large specification or we can say that one test case among large suite of test cases.
- In fact the SCI is a document or the program component like C++ functions or a Java applet.

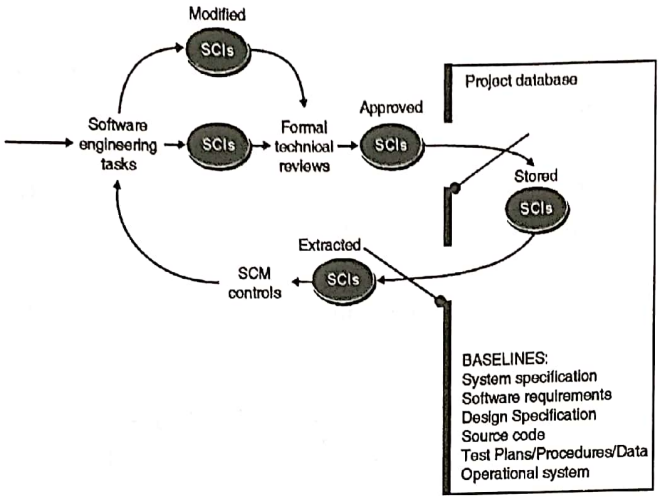

Fig. 11.1.1 : Baselined SCIs and the project database

- Most of the organizations use software tools under configuration control to help development process. In fact the editors, compilers, browsers and various automated tools are the integral part of software configuration.
- In fact the SCIs are catalogued in the project database with a single name and they form configuration objects. These objects are configuration object and it has a name, attribute and it has certain relationship with other configuration objects.
- Referring to Fig. 11.1.2, the configuration objects, Design Specification, Data Model, Component N, Source Code and Test Specification are each defined separately.

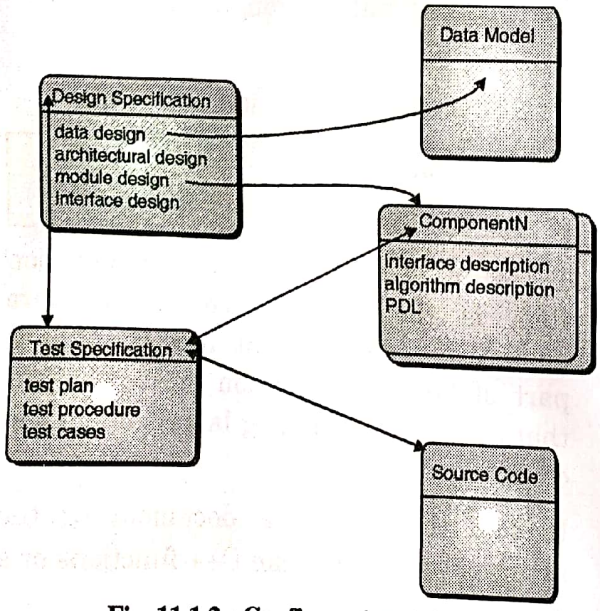

Fig. 11.1.2 : Configuration objects

Syllabus Topic : The SCM Repository

11.2 The SCM Repository

SPPU - May 12, Dec. 14

University Questions

Q. What are the contents of SCM repository ? **(May 2012, 4 Marks)**

Q. Explain the following in brief – SCM Repository. **(Dec. 2014, 4 Marks)**

Review Question

Q. What is SCM Repository ?

- In the early days of software engineering, software configuration items were maintained as paper documents (or punched computer cards), placed in file folders or three-ring binders, and stored in metal cabinets.
- This approach was problematic for many reasons: Many Problems.
  - To find a SCI is a difficult task when it is needed.
  - To determine which items were changed and by whom. It is always challenging.
  - To develop a new version from an existing program is prone to errors and time consuming too.
  - To describe detailed relationship is actually impossible.
- As discussed earlier that SCIs are catalogued in the project database with a single name, they reside in the repository. The repository is a database that stores the software engineering information. The software developer or engineer interacts with the repository by using built in tools within repository.

11.2.1 The Role of the Repository

Review Question

Q. Explain functions performed by SCM repository.

The SCM repository is the set of mechanisms and data structures that allow a software team to manage change in an effective manner. The repository will perform all the fundamental operations of database management system and in addition it will perform the following operations :

- **Data integrity :** Data integrity validates all the entries to the repository and make sure that the consistency among various objects intact and takes care of all the modifications takes place. It will also ensure cascading modifications i.e. change in one object causes change in a dependent object also.
- **Information sharing :** It is mechanism for sharing information among various developers and between various tools. These tools manage and control multi-user access to data, and locks or unlock objects to retain its consistency.
- **Tool integration :** Tool integration is a data model that can be used by many software engineering tools to control access to the data, and performs appropriate configuration management functions.
- **Data integration :** Data integration provides database functions that allow various SCM tasks to be performed on one or more SCIs.
- **Document standardization :** Document standardization is an important task for defining the objects. This standardization is a good approach for making software engineering documents.
- **Methodology enforcement :** Methodology enforcement defines an (E-R model) i.e. entity-relationship model available in the databases i.e. repository. This model may be used as a process model for software engineering. It is mandatory that the relationships and objects must define before building the contents of the repository.
- To achieve these functions, the database is used as a meta-model. This meta-model exhibits the information and how this information is stored in the databases i.e. repository. This meta-model also checks data security and integrity.

11.2.2 General Features and Content

- The contents of databases and features of databases are considered from two perspective :
  - What data is to be stored in the databases ?
  - What services are provided by the databases ?
- A detailed breakdown of types of representations, documents, and work products that are stored in the repository is presented in Fig. 11.2.1.

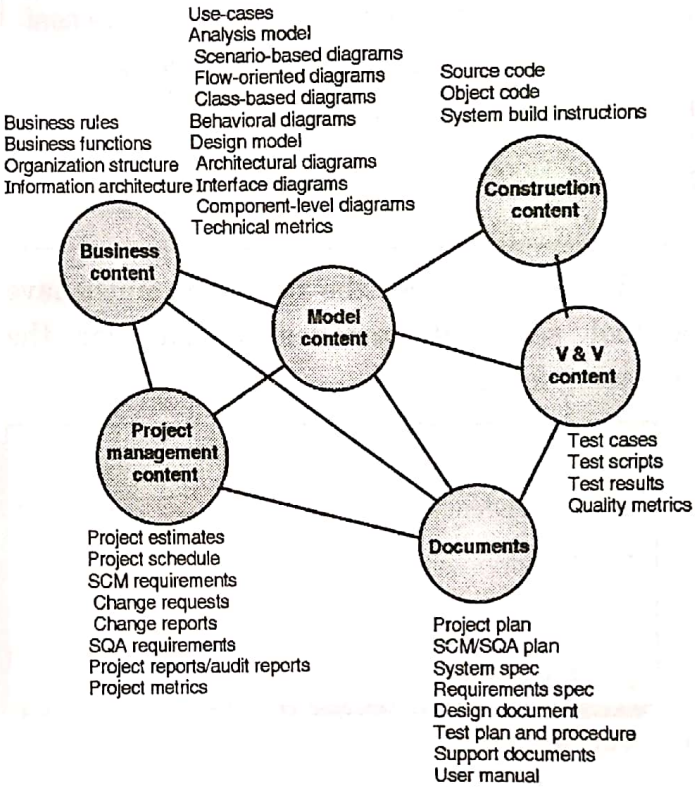

Fig. 11.2.1 : Content of the repository

- A robust repository provides two different classes of services :
  1. The same types of services that might be expected from any sophisticated database management system and

2. Services that is specific to the software engineering environment.

– A repository that serves a software engineering team should :

- Integrate with or directly support process management functions;
- Support specific rules that govern the SCM function and the data maintained within the repository;
- Provide an interface to other software engineering tools; and
- Accommodate storage of sophisticated data objects (e.g., text, graphics, video, audio).

11.2.3 SCM Features

Review Questions

Q. Explain in detail SCM features.

Q. Explain the following repository features with respect to software configuration management.
(i) Versioning (ii) Dependency tracking
(iii) Requirement tracing
(iv) Configuration management
(v) Audit trails.

To support SCM, the repository must have a tool set that provides support for the following features :

1. Versioning
2. Dependency tracking and change management
3. Requirements tracing
4. Configuration management
5. Audit trails

1. Versioning

- In the development process, as the project progresses, various versions of the product will be developed and the database will save all these versions.
- The repository will keep track of these versions in order to make effective management of the product delivery or the product releases.
- The history of all releases will be used by the developers to make effective testing and debugging.

2. Dependency tracking and change management

- The databases or the repository stores various relationships among the configuration objects.
- The relationships may be between entities and processes, or between application design and component design and between all the design elements etc.
- Some relationships are optional and some of the relationships are mandatory relationships that have various dependencies.
- So to keep the track of previous history and relationships is very important for the consistency of the information present in the databases. The new releases of the final product are also dependent on the history stored in the repository. This will be useful in the improvement process.

3. Requirements tracing

- The requirement tracing will provide the ability to trace all the design components and releases. This tracing results from a specific requirement called as forward tracing.
- It will also useful in finding that which requirements are fulfilled properly from the ready product. This is called as backward tracing.

4. Configuration management

- The configuration management is a facility to keep the track of various configurations under development.

- The series of configurations is used as the project milestones and future releases.

5. Audit trails

- The audit trails keeps additional information (i.e. meta-data) like the changes made by whom and when. Also it stores the reasons why changes have been made.
- The source of each change in the development must be stored in the repository.

Syllabus Topic : The SCM Process

11.3 The SCM Process

SPPU - May 12, Dec. 12, May 13, Dec. 13

University Questions

Q. Which are the layers of SCM process ? Explain each in detail. **(May 2012, 6 Marks)**

Q. Write short note on : Software configuration process. **(Dec. 2012, 4 Marks)**

Q. Explain the software configuration management process. **(May 2013, Dec. 2013, 8 Marks)**

Review Question

Q. What is the role of Software Maintenance in Software Product ?

- The SCM (Software Configuration Management) process consists of series of task to monitor the control on changes being occurred. The main objectives of these tasks are as follows :
  1. To identify all individual items that can define software configuration collectively.
  2. Manage the changes taking place in various individual items.
  3. To handle different versions or releases of product.
  4. To maintain the quality of the software under construction over the period of time.
- A process that achieves these objectives must be characterized in a manner that enables a software team to develop answers to a set of complex questions :
  - How does a software team identify the discrete elements of a software configuration?
  - How an organization manages the changes being done in existing release or the existing version? The modification should be incorporated efficiently.
  - How an organization keeps the track and control of new releases?
  - In an organization, who is responsible for authorizing all these changes?
  - The mechanism used to let others know the changes taking place and implemented?
- These questions lead us to the definition of five SCM tasks - identification, version control and change control, configuration auditing, and reporting, as illustrated in Fig. 11.3.1.

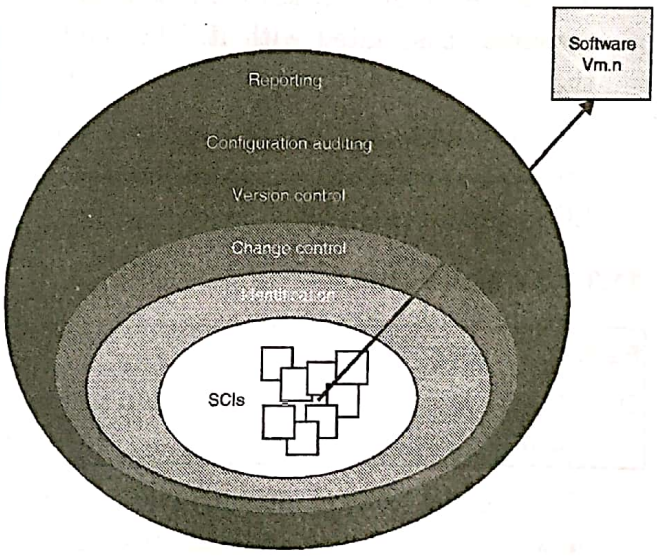

Fig. 11.3.1 : Layers of the SCM process

11.3.1 Identification of Objects in the Software Configuration

Review Question

Q. Write short note on "identification of objects in the software configuration".

- To control and manage software configuration items, each should be separately named and then organized using an object-oriented approach.
- Two types of objects can be identified :

1. Basic objects and
2. Aggregate objects

1. Basic object

A basic object is a unit of information that has been created by a software engineer during analysis, design, code, or test.

2. Aggregate object

An aggregate object is a collection of basic objects and other aggregate objects.

- Since each of the object in the product has some distinct features that make the object different from other objects. The object has a unique name, description and list of resources associated with it. The name of the object should be very clear and distinct.
- The description of the object should identify the type of software configuration item i.e. SCI represented by that object.

11.3.2 Version Control

Review Question

Q. Explain Version control with respect to software configuration management.

- The version control actually controls the new releases or new versions. It combines the procedures and tools in order to control various versions of configuration objects.
- Any version control management system has four major capabilities that is integrated in the version control system itself :

1. The repository will store all the related configuration objects.
2. The repository will store all the versions of configuration objects.
3. It has a facility to provide the related information about configuration objects so that a software engineer will construct a new version based on those information.
4. To track all the issues in development process by using a special tracking facility in the version control.

11.3.3 Change Control

SPPU - Dec. 12, Dec. 13, May 14, May 16

University Questions

Q. Explain the change control process in software configuration process. **(Dec. 2012, 8 Marks)**

Q. Write short note on : Change control process. **(Dec. 2013, May 2014, 4 Marks)**

Q. Explain the change control mechanism in SCM. **(May 2016, 4 Marks)**

Review Question

Q. Explain Change control with respect to software configuration management.

- In a development of a larger software system, the changes may be uncontrolled and it leads to a complex situation. In such projects the change control is done partially by human and partially by some automated tools. In such a complex situation human intervention is very much necessary.
- The change control process is explained in the Fig. 11.3.2.
- The change request is first submitted and then evaluated by a technical support staff by taking into consideration its potential side effects and the overall impact on other objects in the product.

- The other parameters to be evaluated are system functions, the cost of project etc.
- Based on the result of the evaluation treated as a change report, the implementation is taken into consideration. This report is submitted by change control authority i.e. CCA.
- The CCA is a person or a group who has the final authority to take decision on any changes and their priority.
- After approval from CCA, a change order called ECO (engineering change order) is generated for each of the change.
- The object to be changed can be placed in a directory that is controlled solely by the software engineer making the change.
- These version control mechanisms, integrated within the change control process, implement two important elements of change management :
  - Access control and
  - Synchronization control.
- Before an SCI become a baseline, the changes should be applied. The developer will look after whether the changes are justified or not by project. The technical requirement must check properly.
- After approval from CCA, a baseline may be created and change control is implemented.
- Once the final product is released, the formal changes must be instituted as per the Fig. 11.3.2. This formal change is outlined.
- The CCA plays an active role in second and third layers of change control based on size of the project.

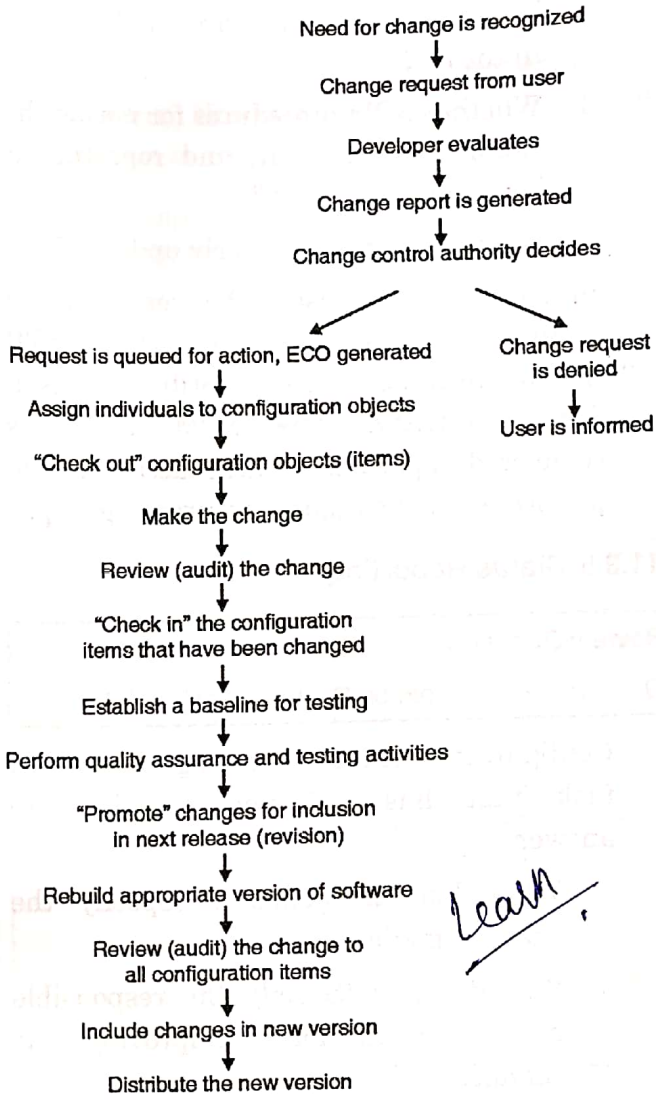

Fig. 11.3.2 : The change control process

11.3.4 Configuration Audit

Review Question

Q. Write short note on Configuration Audit.

- A software configuration audit addressees the following questions:
  - Whether the change mentioned in the ECO applied or not ? Incorporated any additional modifications ?
  - Whether the formal technical review conducted or not to assess technical correctness ?
  - Whether the software process followed or not and software engineering standards properly applied ?

- Whether the changes are "highlighted" in the SCI ?
- Whether SCM procedures for noting the change, recording it, and reporting it been followed or not ?
- Whether all SCIs properly updated ?

– In some of the cases, the configuration audit questions are asked as part of a FTR (formal technical review). Still SCM is a formal activity. The SCM audit is conducted separately. These activities are performed by the quality assurance group.

11.3.5 Status Reporting

Review Question

Q. Write short note on Status Reporting.

– Configuration status reporting is a SCM task that has following questions to answer:
  - What had happened ? (Specify the changes made).
  - Who did it ? (Specify the responsible person or authority approving the changes).
  - When did it happen ? (Specify the time of occurrence of the change)
  - Anything else is affected based on the changes made?

– The CSR report is generated on regular basis to keep the developers aware of the changes made and the history of the changes made. It is very much useful in new releases or constructing new versions.

Syllabus Topic : Configuration Management for WebApps

11.4 Configuration Management for WebApps

– All the steps for the Configuration Management must be followed for the WebApps too. In addition following points must be remembered :
  - Implement a versioning control system (VCS), such as SubVersion (SVN), CVS (Concurrent Versions System).
  - Create a build and deployment process, even if it's just a single page.
  - Block or disable FTP access to host servers,
  - Always commit your files, including documents, to the project repository to maintain version control.
  - Make use of version control systems deployment and version mechanisms.
  - All builds and deployments to test, staging, production, or any platform should be built from the repository, create branches where specific modifications unique to a server or web app instance are needed.
  - Choose a versioning control system that your IDE supports either inherently or through an add-on or plug-in.
  - Do not dismiss following a formal development process as an unnecessary burden, even for small scale projects. Most development processes, such as RUP, will scale to your project size and contain invaluable disciplines.
  - Tag release versions within the VCS, including beta or user acceptance testing versions, so a baseline can be established for these releases.
  - Implement a backup procedure for your VCS repositories, preferably off-site, with a documented recovery plan.

Syllabus Topic : Case Study : CVS and Subversion Tools

11.5 CVS and Subversion (SVN) Tools

– CVS (Concurrent Versions System) and SVN (SubVersioN) are two version control tools that are popularly used by teams who are collaborating on a single project.

- These systems allow the collaborators to keep track of the changes that are made and know who is developing which and whether a branch should be applied to the main trunk or not.
- CVS is the much older and it has been the standard collaboration tool for a lot of people. SVN is much newer and it introduces a lot of improvements to address the demands of most people.
- Probably the biggest improvement to SVN is the addition of atomic commits. Atomic commits allow each commit to be applied in full or not at all. This can be quite useful when the server crashes in the middle of a commit.
- With SVN, the commit can be rolled back while CVS could not undo the partial commit. Another addition is the ability to cleanly rename and move the files in the repository. With SVN the files that have been renamed or removed still carry their revision history and metadata.
- CVS is also unable to push any new changes to parent repositories while it can be achieved in SVN with the use of some tools. These features are simply not supported by CVS or were not a part of its initial design and often cause a lot of problems for some people.
- SVN really works faster than CVS. It transmits less information through the network and supports more operations for offline mode. However, there is the reverse of the medal. Speed increasing is achieved basically at the expense of full backup of all work files on your computer.
- One important feature of CVS is that it allows to rollback any commit in the repository, even if this may require some time (each file should be processed independently). On the other hand, SVN does not allow rollback of commit.
- Authors suggest copy good repository state to the end of trunk to overwrite bad commit. However bad commit itself will remain in repository.

11.5.1 Comparison between CVS and Subversion (SVN) Tools

CVS	SVN
CVS is the much older and it has been the standard collaboration tool for a lot of people.	SVN is much newer and it introduces a lot of improvements to address the demands of most people.
Its slower.	SVN really works faster than CVS.
CVS is also unable to push any new changes to parent repositories while it can be achieved in SVN.	New changes can be pushed to parent repositories.
Atomic commits not available.	Biggest improvement to SVN is the addition of atomic commits. Atomic commits allow each commit to be applied in full or not at all. This can be quite useful when the server crashes in the middle of a commit.
One important feature of CVS is that it allows to rollback any commit in the repository, even if this may require some time (each file should be processed independently).	On the other hand, SVN does not allow rollback of commit.
CVS was initially intended for text data storage. That is why storage of other files (Binary, Unicode) is not trivial and requires special information, as well as adjustments on either server or client sides.	SVN manipulates all the file types and does not require your instructions.
Presently CVS is supported everywhere where you might need it.	SVN not yet so widely used, as the result there are places where it support still not implemented.

Syllabus Topic
Case Study : Visual Source Safe from Microsoft and Clear Case

11.6 Visual Source Safe from Microsoft and Clear Case

- In this section, we discuss the procedure to migrate data kept in Microsoft's Visual Source Safe (VSS) repositories into IBM Rational Clear Case repositories.
- Data is kept in VSS under a single directory tree, and all of the project paths start with a forward slash (/). Therefore, the typical project structure in VSS would look like that shown in Fig. 11.6.1.

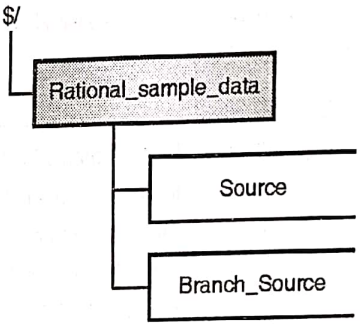

Fig. 11.6.1 : Typical project structure

- Rational_Sample_data is the product name, and it has different directories, which also include directories starting with the word "Branch." These directories are merely branched versions of the existing mainline directories.
- To elaborate, the Source directory is where mainline development takes place, and Branch_Source is the branch development of the Source directory. (This might be confusing if you are familiar with the ClearCase perspective, where you do not have separate physical directories for each branch.)
- The Source directory and the Branch_Source directory may or may not contain different data, depending upon the work being done on both of the development lines. However, the Branch_Source directory will always exist from some version of the mainline development.

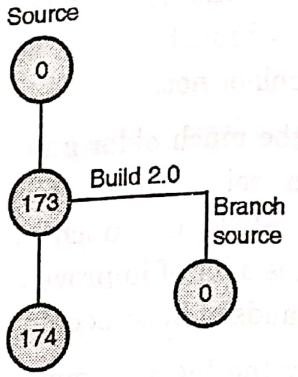

Fig. 11.6.2 : Sample development using branches

- You can use this kind of branching to fix some bugs in the released version of the product, while working in parallel with the development of the next release of the software. In Fig. 11.6.2, a label of Build.20 is applied to Version 173 of the Source directory, and a branch has been created to fix bugs on the Build 20-labeled version.
- Following are the steps involved in export the data from VSS and his can be imported in ClearCase format :

Step 1 : Export data from VSS

Perform these steps to export data from VSS:

1. Open a command prompt and ensure that the PATH variable includes the path of your VSS installation directory.
2. Map the VSS repository to a network drive, using proper authentication.
3. Set the following environment variables:
   - set SSDIR=V: \ {mapped drive of the VSS repository}
   - set TMP=c:\temp
   - set SSUSER=<valid user name who has access to vss repository>
   - set SSPWD = <VSS password of above user-id>
   - Set the VSS project directory to this folder, which has to be exported: ss cp "$/Article/DemoArticle1.0/Source"

- Verify the current project using this command: ss cp

4. Export the VSS directories into a text file by using this command:

```
clearexport_ssafe –r –o
c:\datafiles\vss_export_Source.txt
```

- There are certain switches available with the clearexport_ssafe command, which you can use here according to the requirements:
- –p <date-time>: Process only versions that have been modified since date-time was specified with new metadata.
- –s <date-time>: Process only versions that have been modified since date-time was specified.
- –l <date-time>: Process Important versions only, but include versions created since that specified time. A version is important only if:
- It is the most recent version
- It has a label

5. To keep the same branching structure from VSS to ClearCase (as exhibited in Fig. 11.6.1), use the following command:

```
Clearexport_ssafe –r –s <date-time> -b
Branch_Source -v Build.20
-o c:\datafiles\vssexport_branch.
```

Where:

- Branch_Source is the Target branch in ClearCase
- Build.20 is the Label from which the Branch_Source branch has to be taken out.
- <date-time> is the Time when the Branch_Source branch was created in VSS, so that it processes only the required version on the branch.

6. If there are no errors, the output file should have the export data after successful completion of the export command.

Step 2 : Import data into ClearCase :

- In this step, you will import the data into ClearCase repositories with the help of the output data file that you created in Step 1.
- Now you can start the import process:

1. Create the base ClearCase VOB CC_Migration_VOB (if it does not exist already).
2. Create the base ClearCase view CC_Migration_View (if it does not exist already).
3. Open the command prompt.
4. Set the view context created in Step 2:

```
cleartool startview ClearCase_Migration_View
net use z:\ \\view\ClearCase_Migration_View
```

5. Browse through the VOB until you find the target location, and then execute the following command:

```
cd ClearCase_Migration_VOB
clearimport c:\datafiles\vss_export_Source.txt
```

❑❑❑