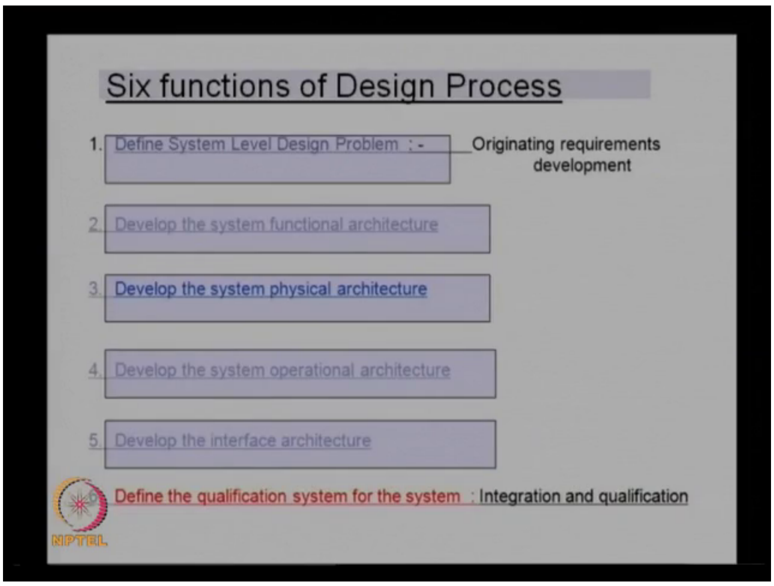
Principles of Engineering System Design
Dr. T. Asokan
Department of Engineering Design
Indian Institute of Technology, Madras

Lecture – 18
System Integration and Qualification

Dear friends, welcome back. We will have another session on the design of engineering systems. In this class we will be looking at the integration and qualification of engineering systems. As I mentioned in the previous lectures this is one of the final stages in the design of engineering systems where we try to integrate different components, different subsystems which we already designed and then qualify the system to make sure that it meets the requirements of the system or it meet needs the stakeholders expectations. The two important points here are the integration and qualification, basically integration is assembly of components and subsystems to make the whole system and qualification is basically the validation and verification of the system to ensure that it meets the design requirements.

So, we will go through these two steps basically the integration and qualification and then see what are the important steps in holding an integration and qualification, as well as what are the procedures to be followed for making sure that the system is integrated and the performance is ensured a through a validation and verification stages.

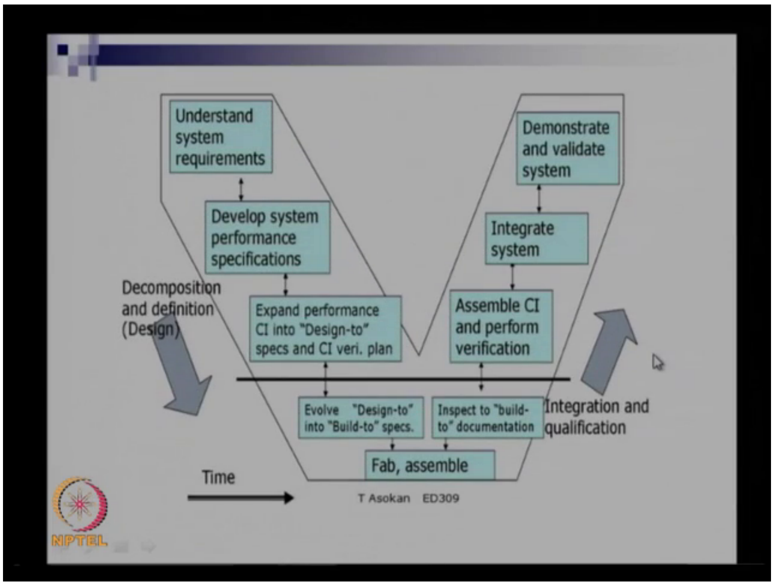
(Refer Slide Time: 01:28)



So, let us go to the design stages which we already discussed that there is a 6 functions of design process and this is the last one which is the defined the qualification system for the system basically we look at the integration and qualification.

So, we are going through these stages that is the system level design problem, system functional architecture, physical architecture, operational architecture and interface architecture. So, we are completed all these stages and the last one the sixth function is basically the qualification system where we look at the integration and qualification aspects of the engineering system.

(Refer Slide Time: 02:03)

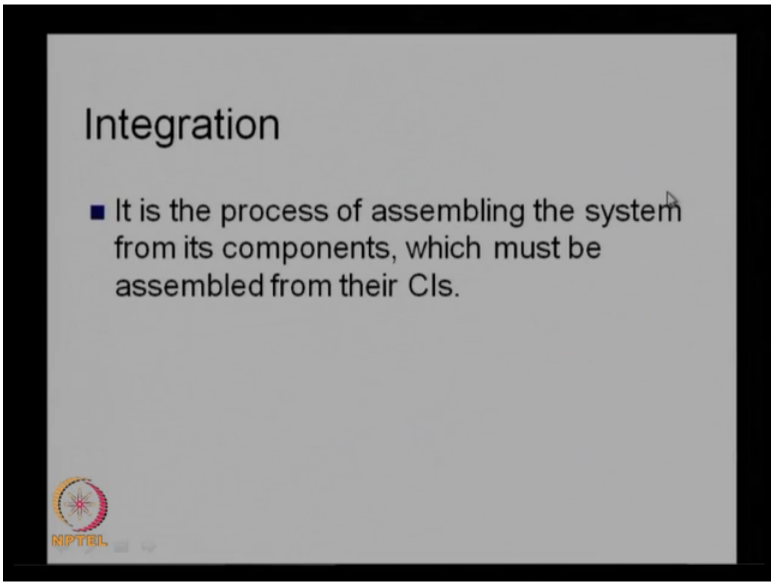


If you look at the system v diagram which we discussed in one of the classes when you discussed the system design process you can see that the system design can be actually be classified into a decomposition stage as well as an integration stage. So, the decomposition stage is basically we try to decompose the problem into small modules and then try to develop these modules and then we start the integration process.

So, here the demon stroke of the v will be decomposing the problem basically we will look at the configuration items and then make them to design to specifications and then verification plans. And this is the part of the system engineers job and then this is the design stage where the design specialists or mechanical design or electronics whatever the specialization needed they will be doing the design stage over here and then once the items are built and we will go for the integration and qualification stage.

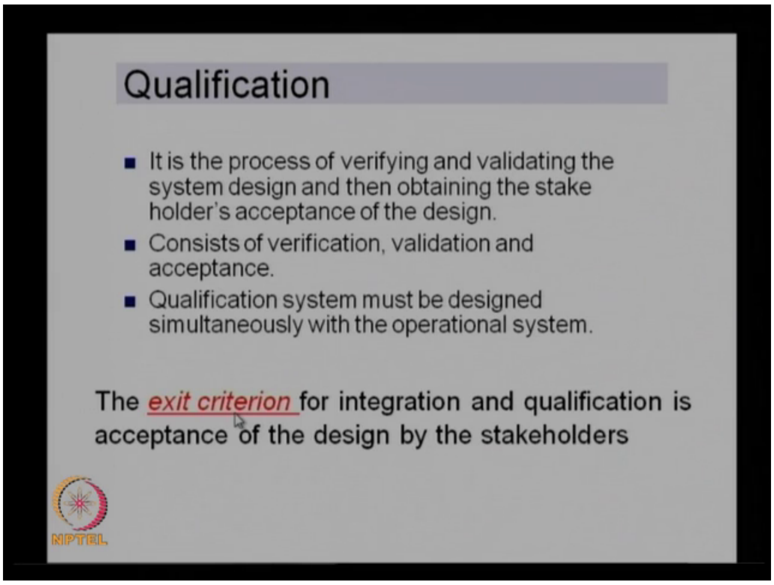
.So, this is the stage where we are now will be trying to integrate the items what are the design and then it will be doing the process of verification and validation you know to qualify the system to make the customer requirements.

(Refer Slide Time: 03:15)



So, the integration basically can be a defined as the process of assembling the system from its components which must be assembled from there CIs. So, it is a process of assembling the components or the configuration items and making them sub assemblies and then making assembly and finally, getting the whole system ready. So, that is the process of integration.

(Refer Slide Time: 03:38)



Qualification on the other hand is basically ensuring that the system actually meets the customer requirements. So, here it is the process of verifying and validating the system design and then obtaining the stakeholders acceptance of the design. So, here this is one of the crucial stages because these are these stage we get the acceptance from the stake holder.

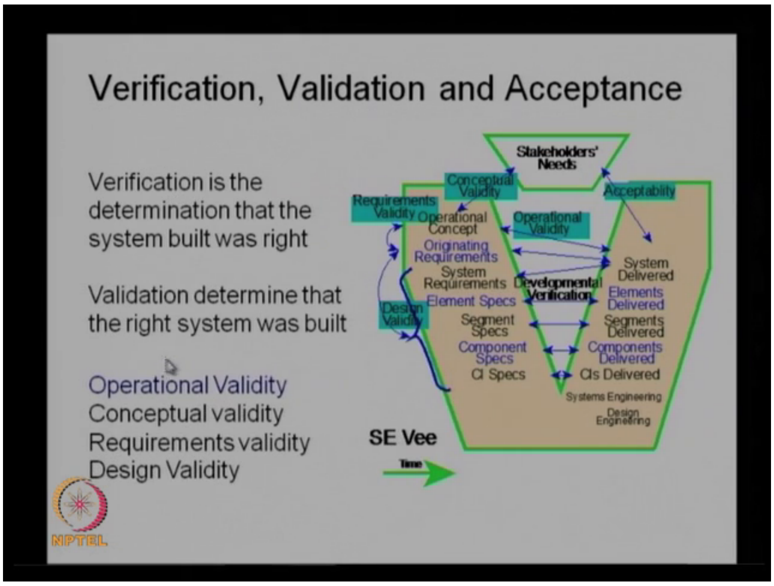
So, once the stake holder says that it is acceptable to them then only we can actually assure or we can actually say that the system design is complete. So, at this stage we try to look at the customer what are the requirements the customer has specified and whether we are actually satisfying these requirements. So, we will go through validation and verification stage and then ensure that it is the system requirements are met the stakeholders requirements are met by the system and the customers and the stakeholders accept the system. So, that acceptance is the final stage of a system design. So, the qualification is the stage where we ensure that the customer is happy with the product or the system and he accepts whatever the functions we provided and that is sufficient to meet the customer requirement.

So, you can see there are two stages one is the verification actually we can see the verification and validation of the system. So, we need to verify and validate the system. So, what is the difference between this verification and validation? Basically a qualification system must be designed simultaneously. So, before going to the difference between verification validation will just me give some important points like the qualification system must be designed simultaneously with the operational systems. So, when we are defining the qualification system that is even before we come to the stage of qualifying the system we need to have the qualification system designed.

So, we need to ensure that we have a particular procedure to make sure that the system requirements are met and the qualification how do we actually do the verification validation process, what are the documents needed, all those seems to be decided at this stage the qualification system development. And this has to be done along with the operational system development because in the operational system we know how the system is going to be operated. So, when we develop the operation system we ensure that we develop a qualification system also which actually details the procedure to be followed in ensuring the quality of the system or in verification and validation of the system.

The exit criterion for integration and qualification is acceptance of the design by stakeholders. So, that is the exit criterion. Once the system is accepted then the integration qualification procedure is over, till the stake holder accepts the product or the system. So, we will not be able to exit from the design process. So, now, to exit criterion here is the acceptance of the system by the stakeholders.

(Refer Slide Time: 06:23)



As I mentioned that verification and validation and acceptance are the three procedures in the qualification process. So, as you can see here the verification is the determination that the system will do us right. So, here we ensure that the system will do us right verification is basically ensure that the system built was right or the system what we are building is the right one what actually we wanted to decide and validation determines that the right system was built. So, that is the difference between verification and validation, validation determines that the right system was built. So, we have some requirements from the customer and customers have specified that the system must meet these requirements and we design a system.

So, first thing is basically to we need to ensure that whatever the system we are designing is actually meeting the requirement or the design is the right system is designed for that particular job. So, that is the validation. In validation we ensure that whatever we are building can actually satisfy the customer requirements and the verification stage we ensure that whatever we wanted to design the same design has been built for the system. That is the that the system built to us right in verification we ensure that the system built to us right and validation we ensure that the right system was built for the particular application. So, this is the basic difference between verification and validation.

As you can see here in validation there are different kinds of validity that is one is the operation validity, other one is the conceptual validity and third one is requirements validity and the design validity. So, in order to ensure that the validation process is complete we need to ensure that all these validity that is the operation validity, conceptual validity, requirements validity and design validity are satisfied. These are basically used to ensure that the right system was built to meet the customer requirements. As you can see here in this v diagram we have to have different validities as well as the verification procedures. I will show you a better picture here you can see the different kinds of validities. So, this is the stakeholder requirement and this is the overall v diagram for that is system design. So, if you look at you can see the different processes involved in the system design. So, these are the system engineering activities and here it is the design engineering activity.

Now, here if you look at the validities the first one is the conceptual validity. So, validity is basically to ensure that the right system was built we have some stake holder needs already identified at this initial stage and then based on these stakeholders needs we develop an operational concepts. So, the conceptual validity is basically to assure that or to ensure that the operational concept whatever we develop actually needs the stakeholders needs. So, basically it is a one to one correspondence or by looking at whether it actually matches the stakeholders needs and the conceptual validity we look at the operational concept and the stakeholders needs and ensure that this particular operation concept can satisfy the stakeholders needs that is the conceptual validity. We need to ensure this conceptual validity before we go for any other processes. So, the conceptual validity is important because if you make it mistake in the conceptual validity we may be making a system and even the verification process it satisfies you would not be able to satisfy the stakeholders needs because the system built was not right.

So, conceptual validity ensures that the operational concept what we chosen for the stakeholders needs to satisfy the stakeholders needs is the right one and the it is valid for that particular needs of the customer. That is the conceptual validity. Then we have the operational requirements. So, we have many requirement from the operational concept and from other sources has been discussed in the initial stages the originating requirements documents will be developed. This originating requirement documents and the requirements identified in originating requirements and the operational concept and if we find that there is a match between these two then that is the requirements validity.

Requirements validity is basically ensuring that the originating requirements are sufficient and the operational concept for this operational concept we identified all the originating requirement. The another requirements have been identified or we have to add few more requirements or there are any requirement which are not needed. So, that kind of an analysis is actually the requirements validation. In this requirement validation we look at all the requirements identified for this operational concept as well as to meet the customer needs and if they are satisfying then we are getting the requirements validity.

Then we have based on this originate requirement then the system requirements we start developing the functional architecture then the physical architecture and then the specification for the components for or the system. So, this actually ensures the descent validity. So, if the component whatever we need to design or the subsystem whatever we design if that can meet the originating requirements and that is the design validity and.

So, in design validation you look at the components and the specification and then ensure that that can these components can meet the requirements already identified. So, that is the design validation. So, at this stage, we have this three validation basically the a conceptual validity, requirements validity and design validity and the last one what we need to ensure is the operational validity.

So, operational validity can be assured only when we have the complete system developed. Once you have integrate all these elements and then get the complete system then we look at the system and its operation and see whether the operational concept whatever we proposed for this customer needs where this is matching with the system what we developed. This is a validated, we can ensure that the operational concept whatever we wanted to develop is actually developed in the original system and that ensures the operational validity of the system. So, these four validation processes the conceptual validation, the requirements validation, design validation and operational validation ensures the validation process of the system.

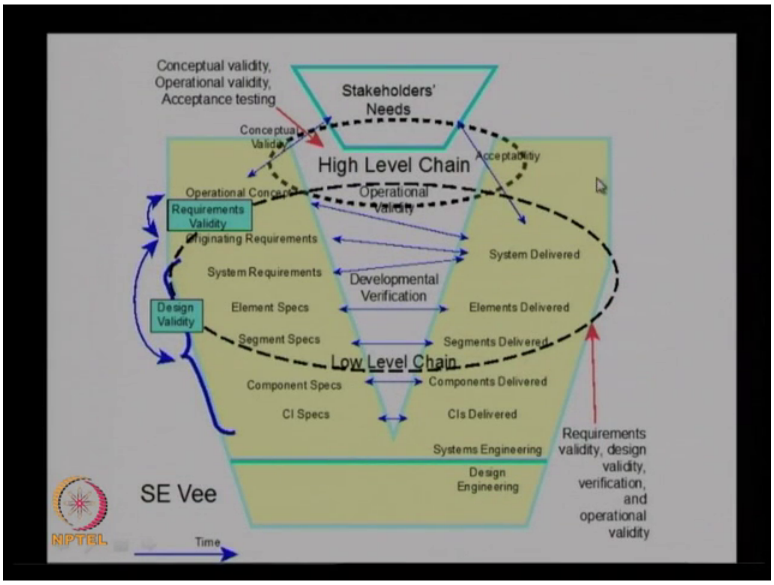
So, that is the first one in qualification, so the basic requirement or the basic idea of validation is to ensure that we build the right system and that system can actually meet the requirements of the customer and this is insured through the four processes of conceptual validation, requirements validation, design validation and operational validation.

And the other one is the verification. So, we have the validation and verification. Verification is basically to ensure that whatever the system we designed and whatever we developed they are actually matching with each other or the requirement or the design requirements or the design whatever we proposed here is actually achieved in the actual design.

So, that is the verification of the system. So, as we develop the system we verify that the system the elements we developed or the sub system we developed actually meeting the requirements of the system we already designed. So, that is the verification. So, ensure that the system will to us right for this, whatever we identified and whatever we delivered are actually matching. So, that is the developmental verification. So, once we have this validation and verification this actually ensures that whatever we wanted to develop that is achieved in terms of the validation as well as the verification and then the last one is basically acceptability of the system.

So, acceptability is basically the process by which the customer or the stakeholder accept the system what we designed. So, once you have these validation and verification it ensures that the whatever we wanted to develop as the design engineers we have developed based on the operational concept and then other requirements and the system is delivered in hardware or software form. And then we do acceptability tests with the customer or the stakeholders and once the stakeholders requirements are satisfied and they will accept the system and that actually completes the cycle of v cycle or we start with the stakeholders needs and with satisfying the stakeholders needs through a design we process where we go through the design process and then we go through the integration and qualification stages and finally, deliver the system as per the customers requirement. So, this is the process of validation verification and acceptance of the system.

(Refer Slide Time: 14:55)

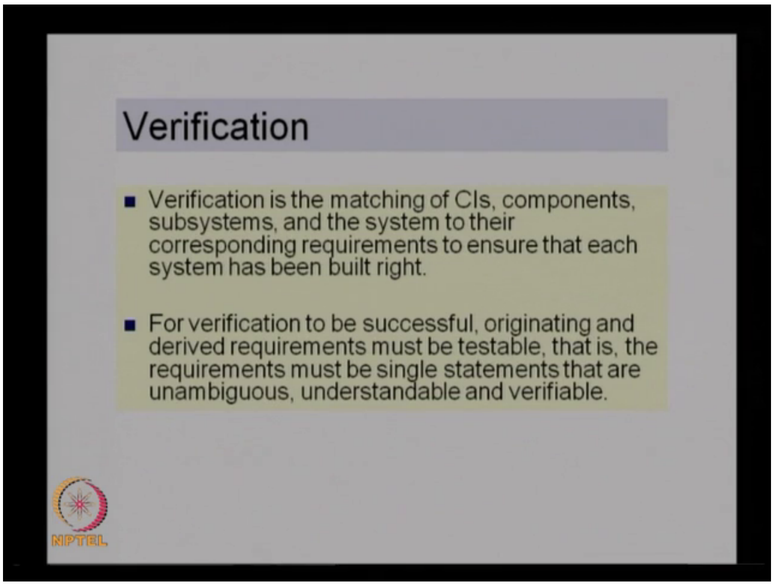


Here you can see in this acceptance the validation and verification there are two chains as you can see there is a high level chain and a low level chain. In the high level chain we look at the conceptual validity operation validity and the acceptance testing. So, the high level this at the top level will be having the conceptual validity which actually ensures that the operational concept can meet the stakeholder needs and the operational validity the operation of the system is as per the operational concept what we proposed and acceptance test is basically the customer accepts the system at what are we designs. So, this is the high level chain which actually as part of the qualification process.

And in the lower level a chain we have the validation the requirement validity, design validity, then we have the verification and operation validity as part of the operation validity it is also coming under the low level chain where we look at the components or subsystems and that their operational validity. That is why we have an overlap of the operational validity in high level chain and the lower level chain.

The lower level chain we may be developing some components or subsystems and then be checking whether actually that meets the requirement that is the operational validity at the initial stages. And then the final stages, once you have assembly of the whole system the complete system is developed then we have the operational validity at the high level chain. So, this is the high level chain and the lower level chain of qualification process which ensures that the tied system has been built and the system built was right. So, this is the process of qualification of an engineering system. I think this is same a diagram which study the more clarity.

(Refer Slide Time: 16:36)



So, let us go through the details of verification and what are the important stages to be employed for verification as we saw that validation there are different kinds of validation. In verification also we need to go through few steps in order to ensure that it meets the system requirement. So, basically verification is the matching of configuration items, components or subsystems and the system to their corresponding requirements to ensure that each system has been built right.

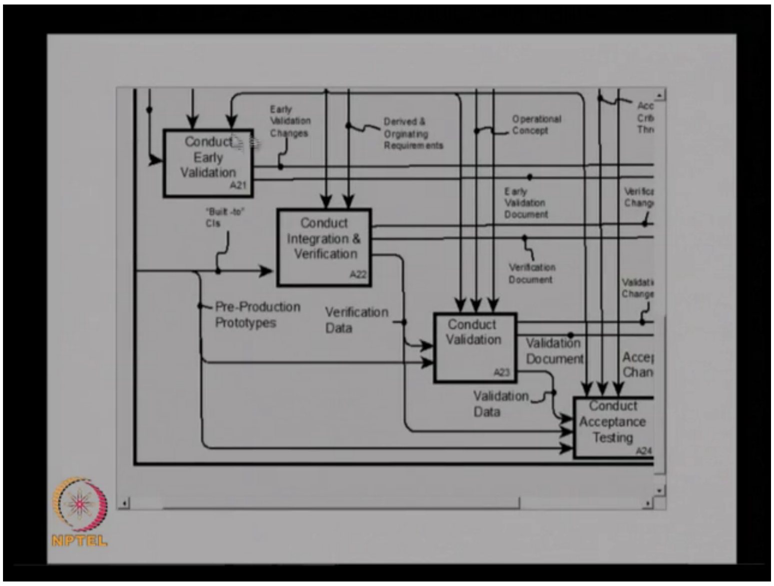
So, in verification stage we look at the configuration items or the smallest level of components and then the component subsystems and the system to their corresponding requirements to ensure that each system has been built right. So, that is the verification stage. We ensure that each system has been built right to meet the customer requirements or the design requirements. And for verification to be successful originating and derived requirements must be testable that is the requirements must be single statements that are unambiguous, understandable and verifiable.

So, this is an important statement for when we start for the verification or the system qualification process we need to ensure that we have all the requirements specified in unambiguous manner. So, when we discussed about the originating requirements development we stated that there are some specific things to be taken care of while we develop the requirements, one of the important points is that the requirement should be very clear and unambiguous that is we should clearly state what actually the requirement from each component or place system and then only we can actually verify whether it is it meets the requirement or not.

When we start the developing the system that is the importance of requirement analysis and requirement documentation because that is going to be used as a statement for verification of the system that is why we need to write it very clearly unambiguously because those who are doing the verification process may not be the same person who actually wrote the requirements. So, it should not be interpreted in a different way that is why all the documentation of requirements especially the requirement from different components and systems the interpreted requirements all those things should be clearly stated and the verification requirements also should be clearly stated, so that this can actually every component every item can be verified against those requirements identified. So, that is the importance of staging the requirements in a clear way.

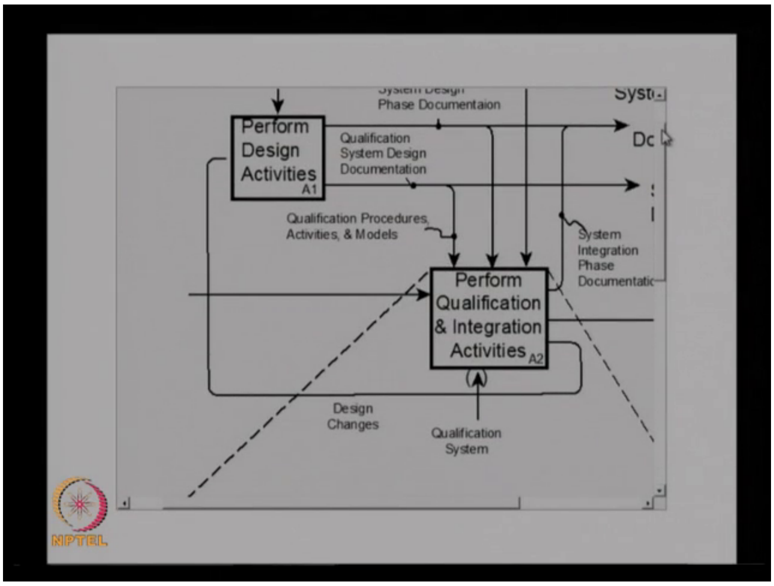
So, it to be successful everything should be single statements that are unambiguous understandable and verifiable. So, this plays a major role in verification. So, identifying the requirements and then writing down the requirements are very important when it comes to the verification of the requirements or the system verification.

(Refer Slide Time: 19:25)



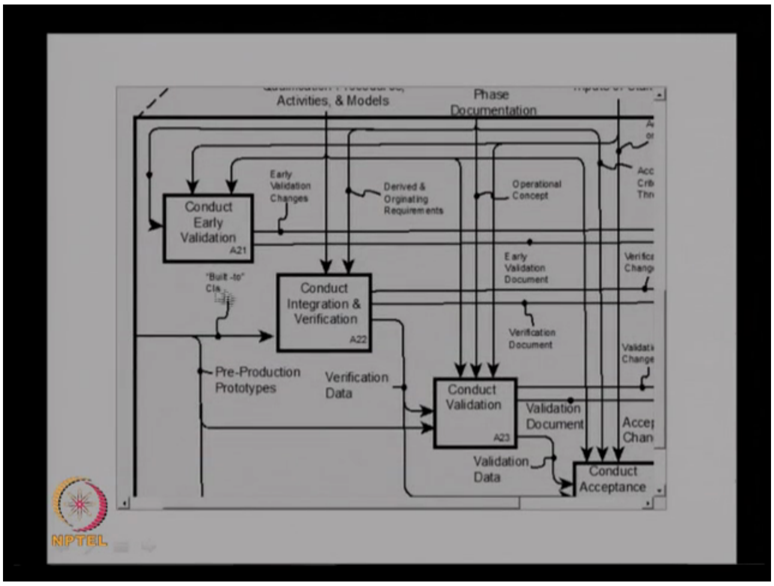
So, you can see here the head of zero diagram for qualification integration. So, you can see these are some of the functions in the top level functions in the system qualification.

(Refer Slide Time: 19:31)



So, system design this is perform qualification integration activities is one of the sub functions in the design process.

(Refer Slide Time: 19:39)

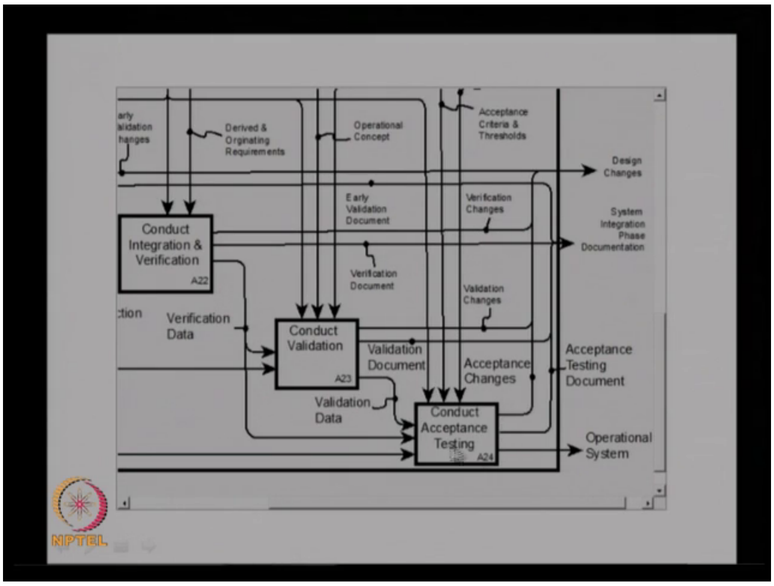


And this can actually be divided into its sub functions as contact early validation. So, the early validation is basically and before we actually start the design of the system we go for the early validation like operation validity to ensure that the operational concept whatever we developed actually meets the requirements and then we go for the integration and verification. So, once we have the other subsystems components designed then we will be having the integration stage and the integration and verification will be carried out and then we will be doing the validation. The final validation where we look at the operational validity and component pressure validity as well as the validation for conception validation and requirement validation will be carried out and then we will be going for the acceptance test. So, these are the various stages involved in integration and qualification of that system.

So, this clearly validation basically ensure that the system or the operational concept whatever be developed is sufficient to make the customer requirements and then from this stage we will go for the design. We can start their designing the system in terms of its functions functional architecture physical architecture and the operation architecture and we develop the configuration items specifications and using those specifications will carry out the integration and verification of the system to ensure that the system will do us right and then the right system was built will to look for the again a validation.

So, we will be getting the validation document will be using the validation document to do this and verification data will be used as a less the prototype and other data will be used for validation purpose. And one, using the validation data and the validation document and other details from the ORD, the originating requirements documents we will be doing an acceptance testing which will ensure that the system designed is accepted by the customer or it actually meets a customer requirements.

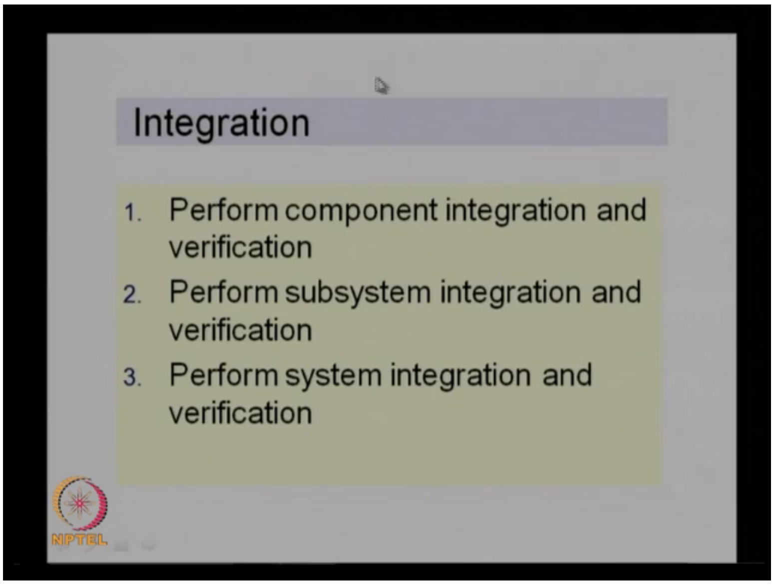
(Refer Slide Time: 21:38)



And once this meeting the requirement then we are getting an operational system meets the requirements of the customer and we will be getting the acceptance and testing document also as part of the output. And the final output from this stage will be basically an integration documentation and acceptance test documentation and you will be getting the operation system also. And whenever there is any design changes that also will be brought into these stages. So, that all the design changes are incorporated in the validation and verification stages and that also will be getting as an output from this a stage of a design process.

That is the various steps involved in the verification and validation in the qualification a system.

(Refer Slide Time: 22:34)

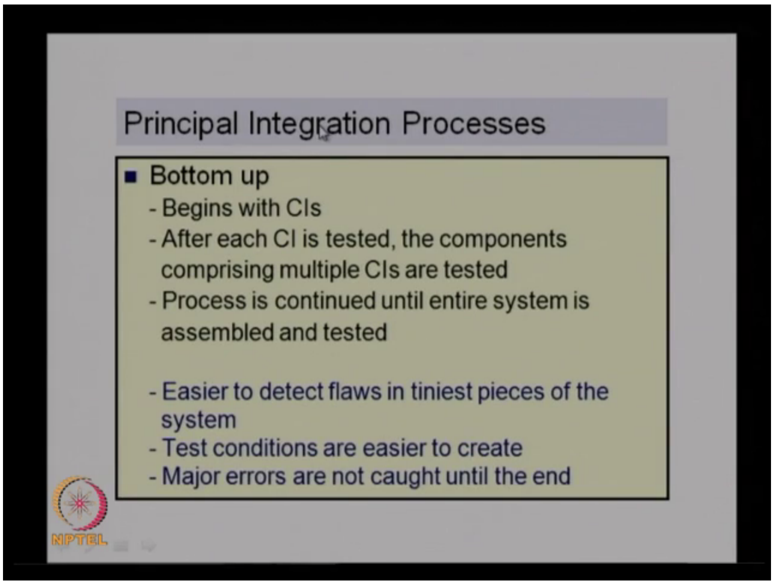


Unless I mentioned we look at the verification and validation of the system and we need to look at the integration of the system or how do we actually do the integration of components in the system. So, there are different ways of doing it, especially when it comes to hardware or software that the methods used are different when compared to the hardware systems. So, we need to see what are the different procedures available different methods adopted by various industries and various organizations and various sectors of organization in developing, in integrating system.

Basically integration is to get the assembly of all the components and through ensure that it actually the when you assemble the components configuration items components and subsystem and get the complete system with performance is ensured to meet the customer requirements. So, here and the integration we do the component integration and verification, we will do integration of the components and then verify whether that meets the design requirements and then we perform the subsystem integration and then do the verification again to ensure that it meets the requirement and then perform system integration and verification. So, these are the three stages normally followed in integration purpose.

First we go for the component integration and then go for the sub system integration and then go for the system integration. So, basically this is more like a bottom up approach where we start with smallest item of the components then go to the sub system and do the system integration. And every stage, we will be going for the verification. So, whenever we have a component integration we go for a verification stage to ensure that this needs the descent requirements and then we go for sub system integration and then go for a verification again ensuring that it meets the requirements and then perform the system integration that is the final system integration and then do a verification to ensure that it meets the requirement specified. So, these are the steps involved in the integration process.

(Refer Slide Time: 24:44)



And there are different processes followed then integration are one is the bottom up approach. As I mentioned in bottom up approach we begin with the configuration items and after each configuration item is tested the components comprising multiple configuration items are tested and the process is continued until entire system is assembled and tested. So, this is the bottom up approach.

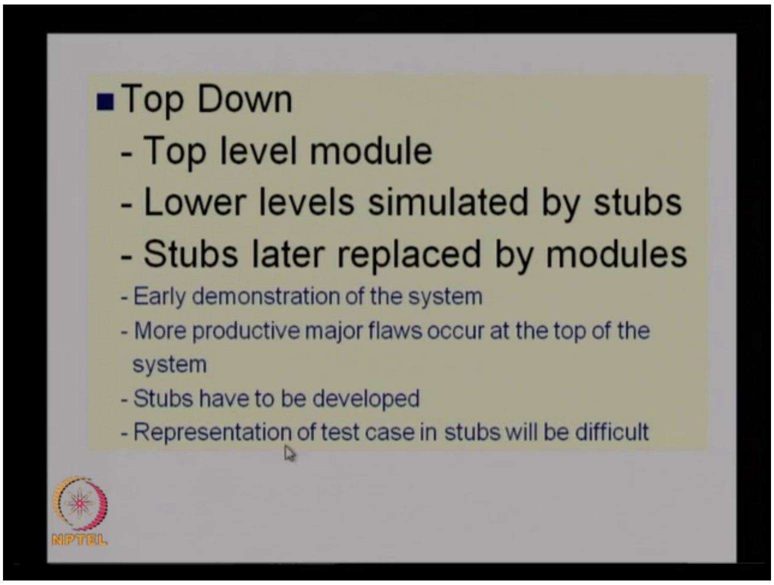
It begins with the configuration items and after each configuration item is tested. So, we do the testing of configuration items initially each CI is tested the components comprising multiple CIs are tested. So, we do the individual testing and then go further components with multiple CIs and we be tested and then it is continued until entire system is assembled and tested. So, it will go one by one they start with the smallest the CI configuration item test each configuration item and then assemble the configuration items to components then do the testing and then assembled the just components till the entire system is assembled and then do the testing. So, that is the most common bottom up approach in integration.

The advantages of this method is basically to easier to detect flaws in tiniest pieces of the system. Since we are starting with the configuration items or the tiniest pieces of the system and we are doing the testing it is easy for us to find out the errors in any small item small pieces of the system. So, that is one of the advantages of using the bottom up approach. And then the test conditions are easy to create because they are testing the individual CIs or the small items very easy to develop the test conditions, unlike testing the whole system will be doing the individual testing of the system, so we can write down the procedures very easily since they are very simple products simple pieces or not so complex components.

The main problem with this is that the major errors are not caught until the end because most of the times there will be errors in the integration and these errors are not found till the last moments. The individual items may be perfect, so we can actually identify the flaws in the individual items, but the major errors of integration will not be caught until the end or towards the end of the design process then only will be able to catch the major errors. So, that is one of the problems with the bottom up approach.

Most of the system design is follow the bottom up approach because of these advantages that we can test small parts. And it is easy to develop the test condition for small parts, but the major problem is there is this one that you do not to get the major error is until the end of the system design or towards the or end of the design process integration process then only you will be getting the major errors and solving those major errors at that stage maybe and may be difficult also.

(Refer Slide Time: 27:42)



And the other method is basically the top down approach, in top down approach the top level modules are chosen and lower levels are simulated by stubs. So, its stubs are later replaced by modules. So, in this case we take the top level module. So, not all the sub modules or sub system will be available, so for this we will use some kind of stubs to just simulate the behavior of the lower level modules and then we test the top level module and if this working then we will the steps will be replaced by the actual module. So, this is the way the top down integration works.

So, for example, if you take a laptop assembly we will be doing it without a hard disk and for a hard disk we can actually simulate the conditions of hard disk using some software or some other methods and then if the whole system works then you replace that step using an a actual module. So, like this you can actually have many stubs in the system which are actually simulated modules we are not the real modules, we use those simulated modules in the system and then test the top level module to ensure that it works well and then one by one these modules and steps will be replaced by the small the actual module. So, that is the process of top down approach.

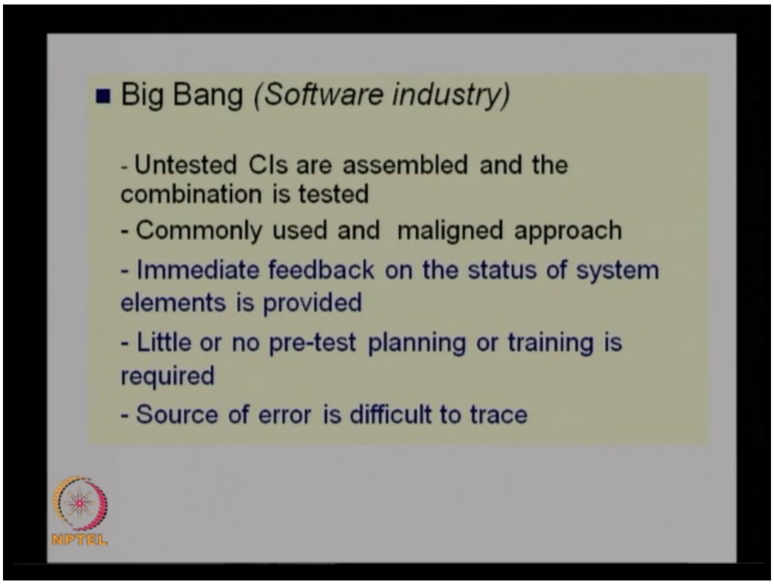
So, here it is possible to do an early demonstration of the system. So, because of the top down approach we can demonstrate the system at an early stage though the modules are not ready they are only stubs inside we can still say that this actually works when we have the whole system built it will be like this and then we can demonstrate at this stage.

But in the case of bottom up approach we will not be able to demonstrate the system until all the modules are assembled and tested. So, that is the advantage in top down is that you can actually demonstrate the behavior at an early stage.

The other one is the more productive major flows occur at the top of the module system that is most of the major flows can be easily identified at the top of the system. So, that it can be identified at the early stage itself because they occur at the top of the system most of the major flows occur at the top level so that can actually be easily identified using in this process. The only disadvantage is that the steps have to be developed at a later stage. So, though we have a demonstration of the system the steps are not ready and the we need to develop the actual step and then any variations in the stub and the simulated the actual module and the simulated one if there is any difference that will affect the performance of the system.

Though, we have an early demonstration of the system that will not ensure that the system will work because the development of stubs, and then the errors in these stubs may affect the performance of the system. So, that is the disadvantage with the top down approach. And then the representation of test case in stubs will be very difficult. So, since we are using the simulated modules here the representation of test case and stubs will be very difficult. Again development of the stubs also be with difficult and then the representation of this test case also will be difficult in the top down approach these are the under legislation as the disadvantages of a top down approach.

(Refer Slide Time: 30:59)



And the other one is known as a big bang approach and this is basically used in the software industry the software industry has got a different way of working. So, they use they do not go for a bottom up approach or a top down approach they basically go for a as big bang approach in this big bang approach the untested CIs are assembled and the combination is tested.

So, they actually start with the CIs, but they do not test the CIs. So, what they do? They take this CIs the configuration items or the smallest the tiniest parts of the system and they are assembled and the combination is tested. So, this is different from the bottom up approach, in bottom up we will take this CI and test the CI and then only we do assembly, but in a big bang I will take the CIs directly assemble them and then do the testing and if it is working just I accept it if it is not working then try to see what is problem. This is a commonly used and, but maligned approach.

So, it is not a very good approach, but it is very widely used in the software industry because of the time constraints in the software industry as well as most of the CIs will be a taken from some other systems or which is used by some other system. So, they feel that it is not necessary to do the testing of all the CIs that basically go for directly assembling the CI and then testing them and then the advantage is that immediate feedback on the status of system elements is provided. So, we can actually get a immediate feedback on the status of the system elements or whether they are working or not otherwise there is any problem with the system or can I where in the CI can be easily identify just by taking, just by testing the combination of CIs.

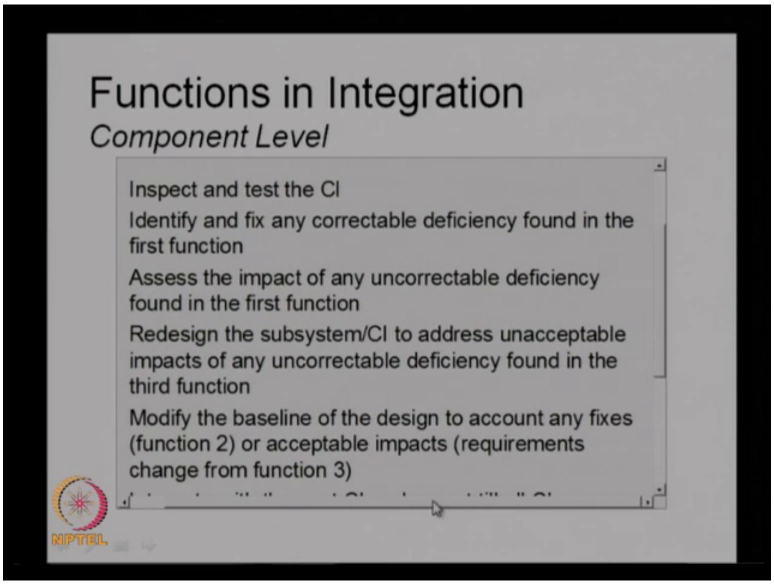
Then little or no pretest planning or training is required. So, since we do not have any testing of the CI there is not much of testing or the planning is needed and you can directly go for combining them and then doing a testing. So, no training or planning is required at this stage. And major problem here is that the source of error is difficult to trace since we are not testing the configuration items or the smallest elements there is no way to find out where actually the error came from. So, if the assembly is not working we will assume that this problem is there in some of the CIs, but it is very difficult to say where actually the error happened. So, that is why the source of error is very difficult to trace in the big bang method of integration.

So, these are the three methods of integration this top down, bottom up and the big bang. Bottom up is the normally employed a strategy where we do the testing of CIs with the smallest part of the component and then assemble them to components and then do the testing and then assemble these components to the system and continue the process until all the components are assembled to get subsystems and then finally, system and do the testing. So, that is the most conventional way of bottom up approach in integration.

Another one top down approach is basically you some steps to simulate the bottom modules. So, take the top level module and then using some steps or the simulated modules test the top level module and if it is working then go for they are developing the actual sub modules, that is the top down approach. And the third one is the big bang approach where I did they only used by the software industry and they will take the CIs without any test and assembled them together and then see whether it is working or not and if it is working they will go for the next level of assembly.

So, here the problem is that you do not get the source of error. So, most of the times it is very difficult to a trace the point where actually the system failed, but they are widely used because of the advantages and the easiness in which we can do the testing without any training or any pretest planning. So, that is why big bang is normally used in software industries and the probably we are getting lot of bugs because of this kind of an approach also.

(Refer Slide Time: 34:58)

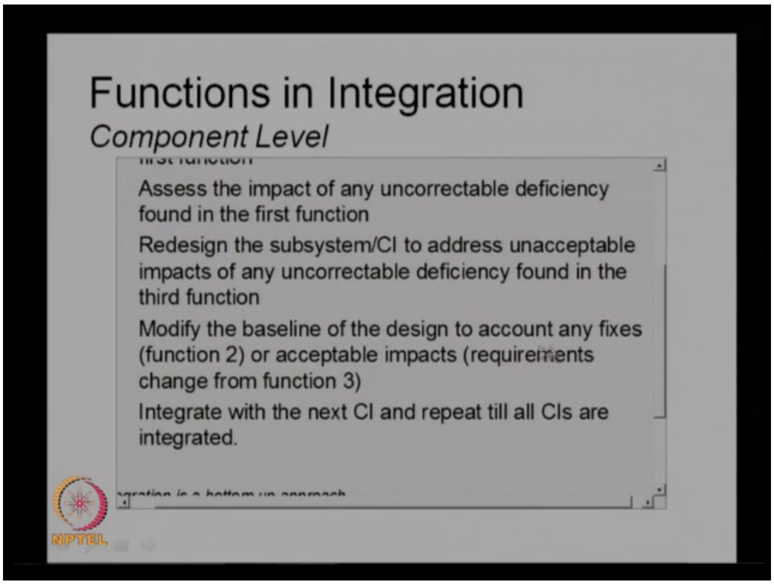


So, these are the main functions to be done during integration. So, in the basically that bottom up approach we inspect and test the configuration item and identify and fix any correctable deficiency found in the first function. So, we take the configuration item and then identify and fix any correctable difference found in the functions first function.

So, when we find any error in this confirmation item we try to fix that particular deficiency and assess the impact of any uncorrectable deficiency found in the first function. So, sometimes when you do the testing of the configuration item we may have find that there are some uncorrectable errors then we look at the impact of this uncorrectable errors and see what actually will happen in the system if we just leave it like that. And then you re-design the CI to address unacceptable impacts of any uncorrectable deficiency found in the fluid function that is if you have some uncorrectable deficiency and we found that the we assess the impact and if there is a serious impact of this uncorrectable deficiency then we go further redesign of the system.

If there is no not much of impact at the impact is very minimal then we do not really need to go for the redesign we will accept that component with the uncorrectable deficiency.

(Refer Slide Time: 36:20)

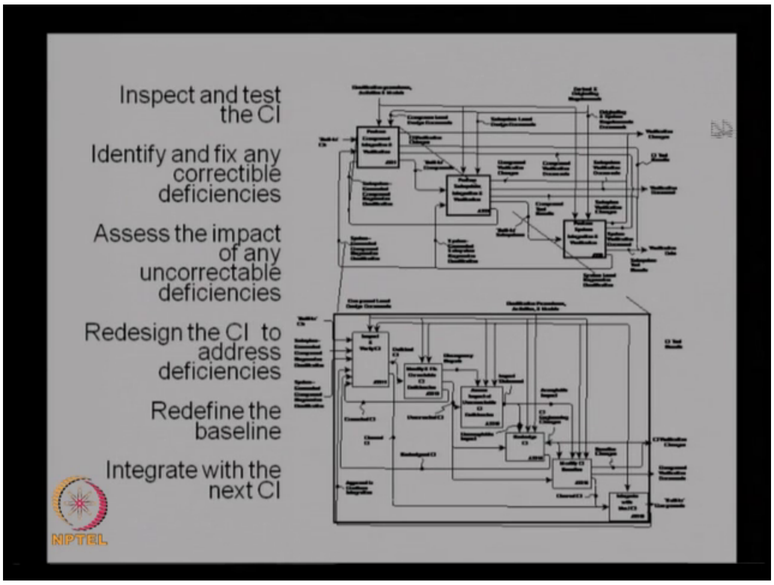


And the modify the baseline of the design to account any fixes or acceptable impacts needed that is if you want to change the baseline of the design that is you already have a baseline of the design with the design requirements and specifications. So, if any fixes are needed because of the previous one that is the uncorrectable deficiency or its impact. So, that is if anything is needed we will change the base baseline of the design to take into account these fixes and the acceptable impacts. And then integrate with the next CI and repeat till all CIs are integrated we will continue with the same process till are the CIs are integrated.

So, in the integration process the component level we start with the individual component or the CIs, do the testing and if it passes will accept it. And if it is not passing the test will try to find out what is the uncorrectable deficiency and what is the impact of this uncorrectable deficiency and if needed we will change the we will redesign the system and then again change the baseline of the design to meet this uncorrectable deficiencies or the fixes we need to make and then continue with this all the till all the CIs are integrated.

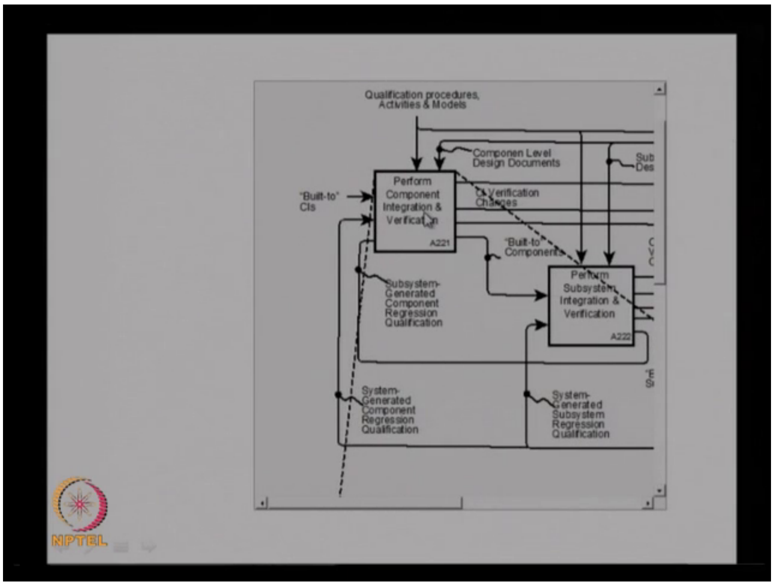
So, that is the functions in integration at the component level integration we need to follow.

(Refer Slide Time: 37:36)



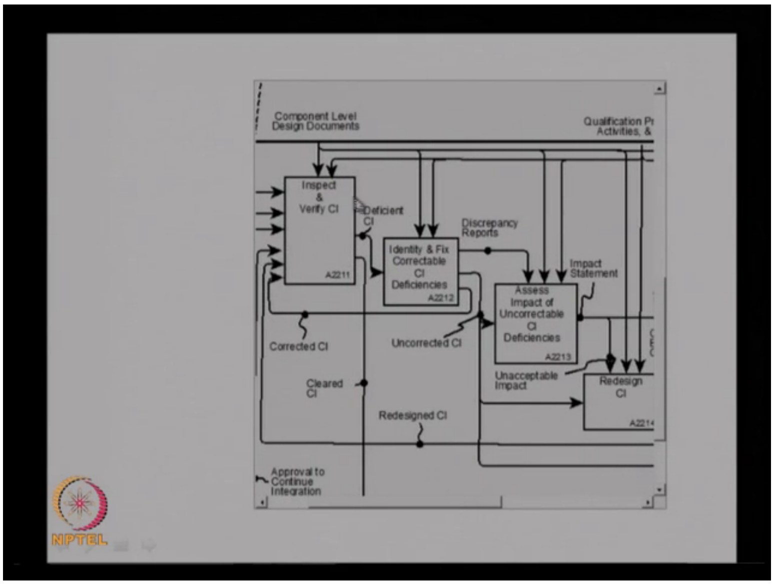
This can actually be represented by a diagram these are the stages basically inspect and test the CI identify and fix any correctable deficiencies, assess the impact of any uncorrectable deficiencies, redesign the CI to address efficiencies and redefine the base line. So, these are the stages involved and with integrate with the next CI. So, this can actually be shown as an id of zero diagram.

(Refer Slide Time: 37:59)



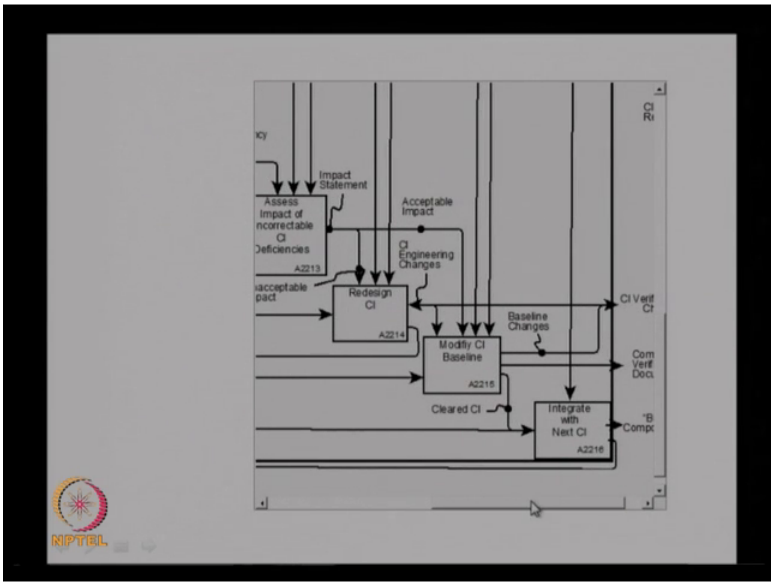
As you can see here, component level verification, this is the perform component level verification is one stage in the system integration.

(Refer Slide Time: 38:11)



So, this can actually be shown as with multiple sub level functions that is inspect and verify CI is one of the functions and then identify and fix correctable CI efficiencies. So, if you have any correctable deficiency that can actually be fixed then assess the impact uncorrectable deficiencies and then redesign the configuration items then modify the baseline and then continue with the next CI.

(Refer Slide Time: 38:37)

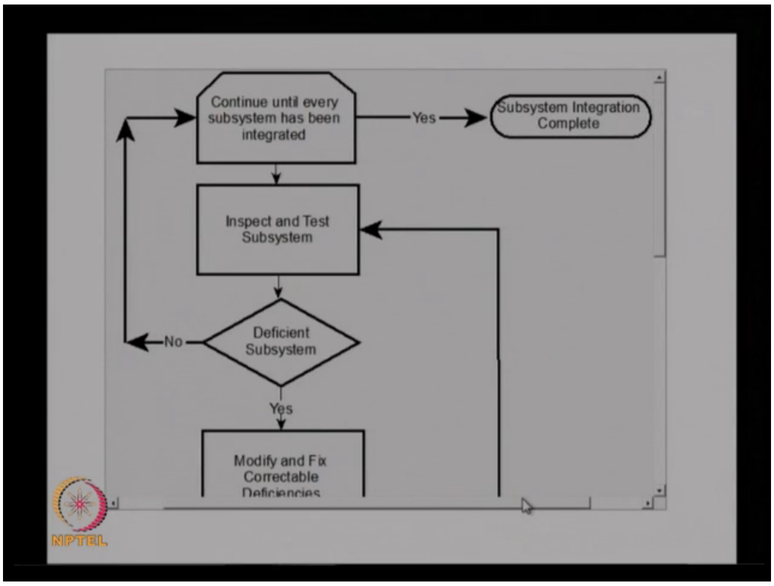


So, these are the sub level functions which can actually be identified in as part of the component level will system integration.

So, this integration and verification stage we will be going through these stages to ensure that each component is tested and it meets the requirement and if it is not meeting the requirement we go for the correction of the component, we have redesigned the component and if needed we change the design base line or the design requirements also and that my baseline modification will be coming as the part of the verification document as well as the design changes. These are the output coming from these particular stage where we have the verification changes the configuration item verification changes will be coming out and that will be used for changing the baseline of the design requirements.

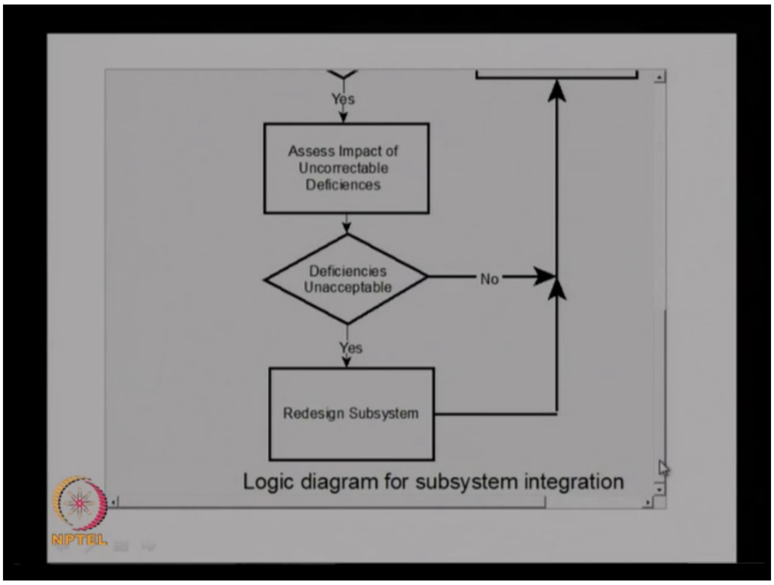
So, these are the stages involved in integration component level integration.

(Refer Slide Time: 39:29)



So, the same thing can be represented as a block diagram here. So, this is basically for testing the subsystem integration, so the component level integration in the previous diagram. So, this is the subsystem level integration.

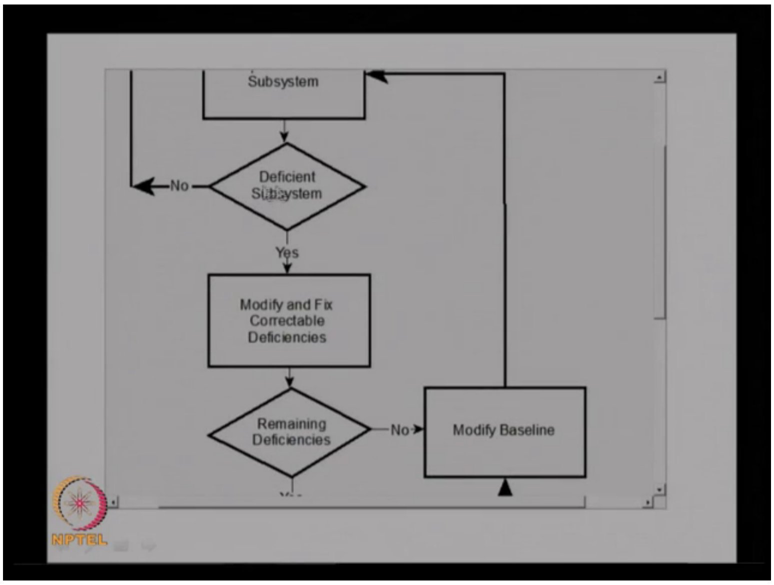
(Refer Slide Time: 39:45)



Once we have the component level integration we need to go for the sub system level integration. In sub system level integration we will start with the subsystem has been integrated. So, will look at the weather all the subsystem have been integrated if not we start with the sub system integration and inspect and test the subsystem and then whether very earliest deficiency in the sub system will be checked, if there is no deficiency it will go a back and then start with the next integration process.

So, like that the previous case is looked only at the component level or CI level that the subsystem level we start the integration, everything is completed then the subsystem integration is completed and if there is any deficiency in the system then this is a deficiency subsystem will modify and fix the correctable deficiencies.

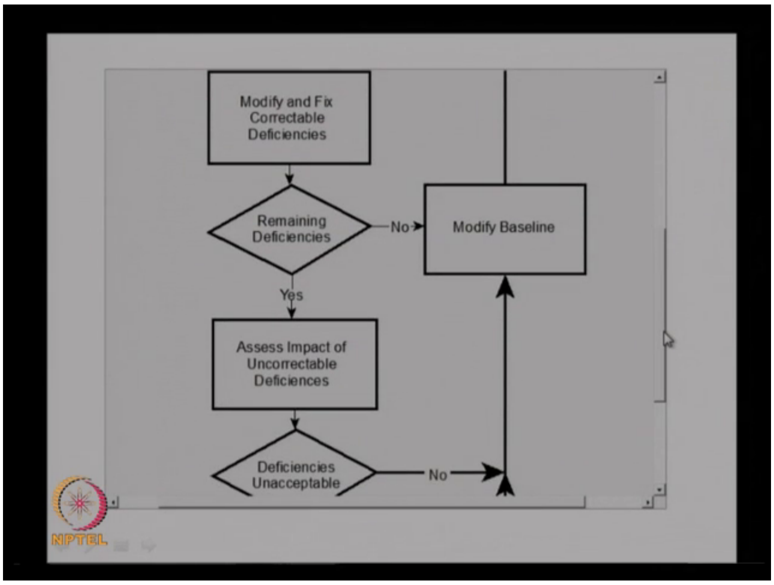
(Refer Slide Time: 40:24)



And then we look at the remaining deficiencies some deficiencies cannot be corrected, there are any deficiencies to be corrected then if there are no deficiencies we will modify the baseline based on what are the fix we made or what are the modification we made that will change the baseline of the design. And if there are still some uncorrectable deficiencies we look at the impact of these uncorrectable deficiencies and then whether there acceptable or not if they are acceptable we go for this if they are not acceptable then they again change the baseline and acceptable deficiencies we redesign the subsystem and if they are not acceptable then we redesign the subsystem and then I again go back to the integration stage.

This is the logic diagram for a subsystem integration similar to the component level integration we start with the subsystems and again test it and then see whether they are actually meeting the requirement or not. They are not making the requirement we either fix the descent the flaws if cannot be fix we will go for the assessment of the impact.

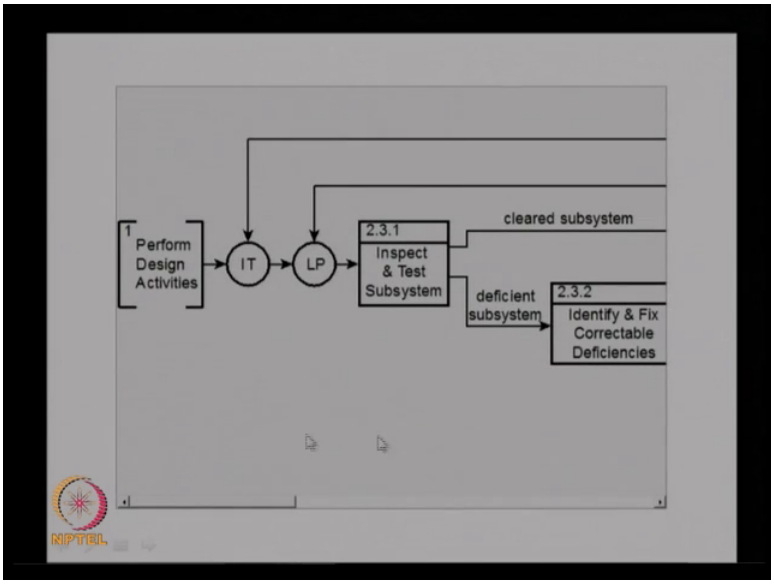
(Refer Slide Time: 41:30)



And if the assessment of impact is acceptable then we take it otherwise if it is not acceptable then we go for the redesign of the subsystem and then again go for integration inspection and test and then check the system. This will be continued till all the subsystem has been integrated and once it is completed then the subsystem integration process is complete.

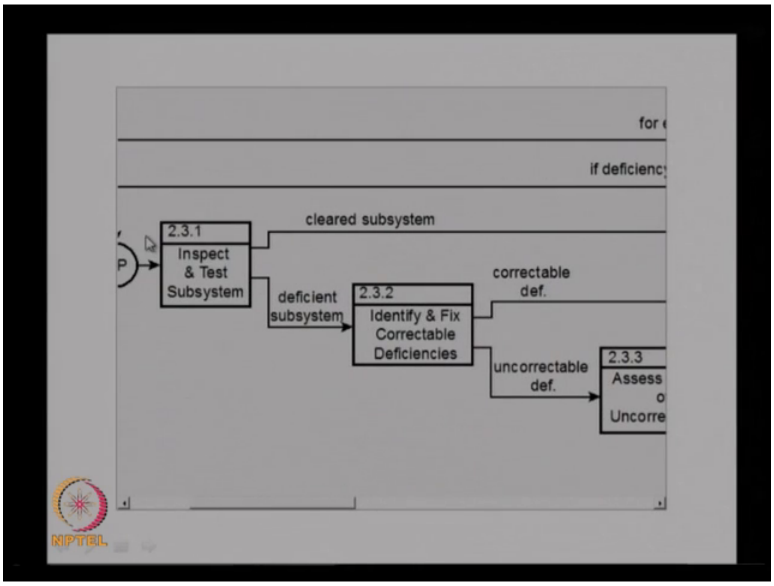
So, this is the way how the sub system integration is being done. So, the configuration item integration or the component level integration and subsystem level integrations. And the same thing can be actually be shown as a functional flow block diagram.

(Refer Slide Time: 42:00)



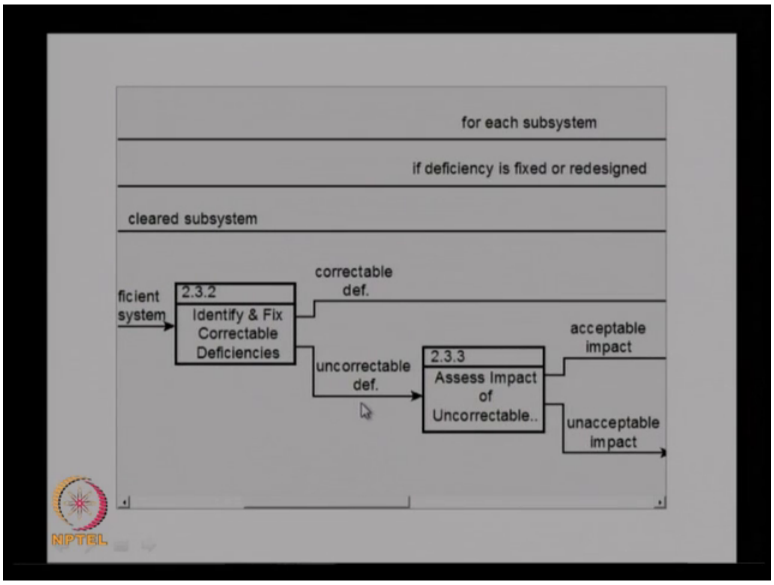
We will discuss about the function block diagram in one of the coming lectures. So, this basically the same flow chart whatever we a shown is shown as here as a function flow block diagram it represent the iterations and LP is a loop. So, we perform the design activities and then inspect and test the subsystem and the deficient subsystem will go through one channel.

(Refer Slide Time: 42:26)



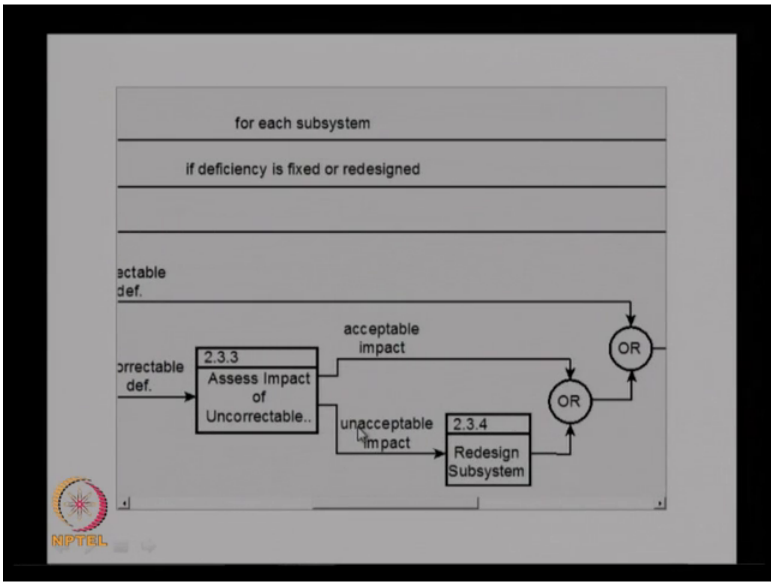
And clear subsystem will go through another one and it will be unacceptable deficiency and correctable deficiencies and here again acceptable impact and unacceptable impact.

(Refer Slide Time: 42:30)



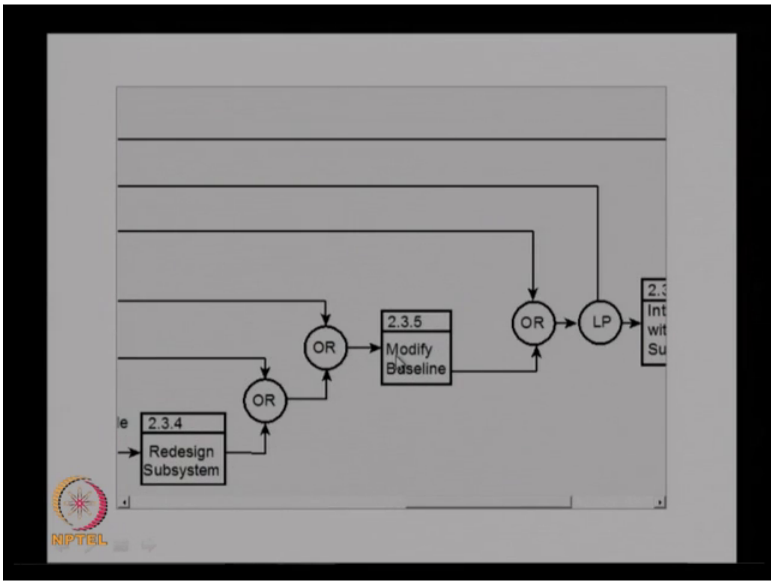
So, there is an unacceptable impact we redesign the subsystem and it will go through to the modified baseline.

(Refer Slide Time: 42:34)



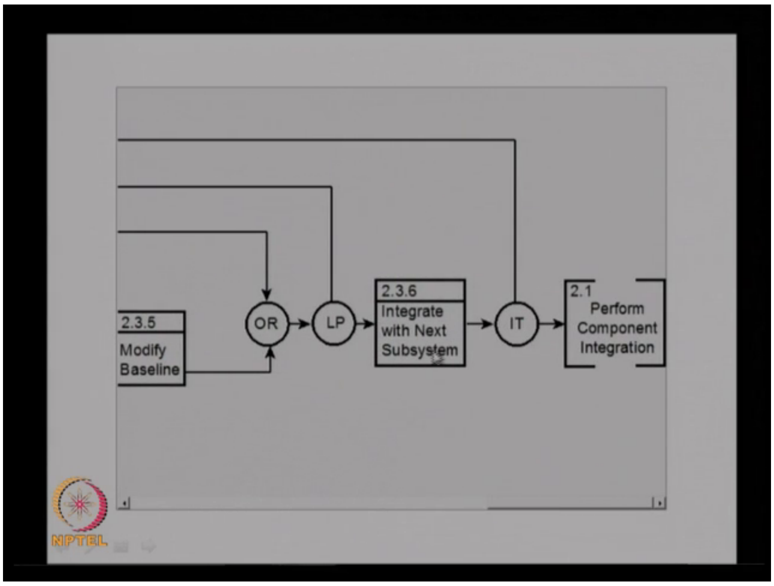
And then again go out here and go back to the testing stage from here it will go to the testing again.

(Refer Slide Time: 42:42)



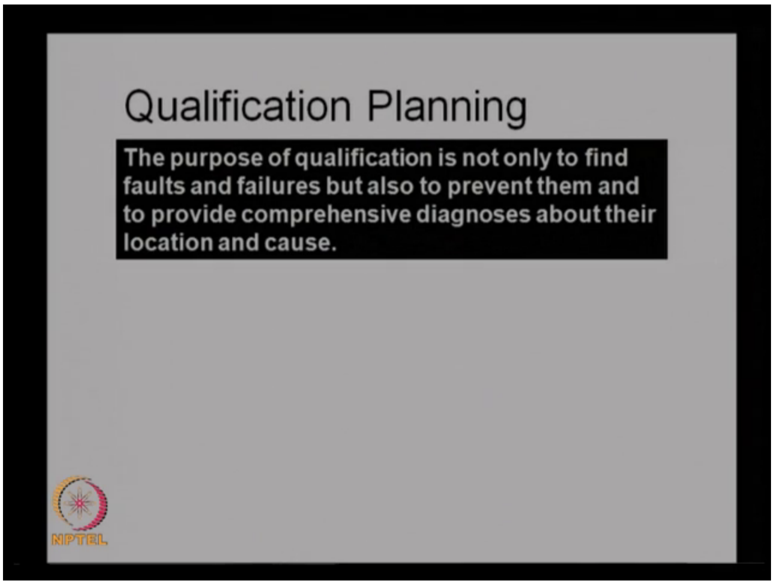
And then once it is tested and cleared it will come to the integration with in next subsystem and continue its till all the integration has been completed.

(Refer Slide Time: 42:55)



So, this is the process how we do the subsystem level integration.

(Refer Slide Time: 43:02)



The next one is qualification planning we will see in the next lecture. Basically what we had completed the integration procedures and how do we do the integration, the next one is how do we actually do the validation and verification, what are the procedures, what kind of methods can be used for verification, what are the testing methods to be employed for acceptance. So, all these things we will be seeing in the next lecture, till then good bye.