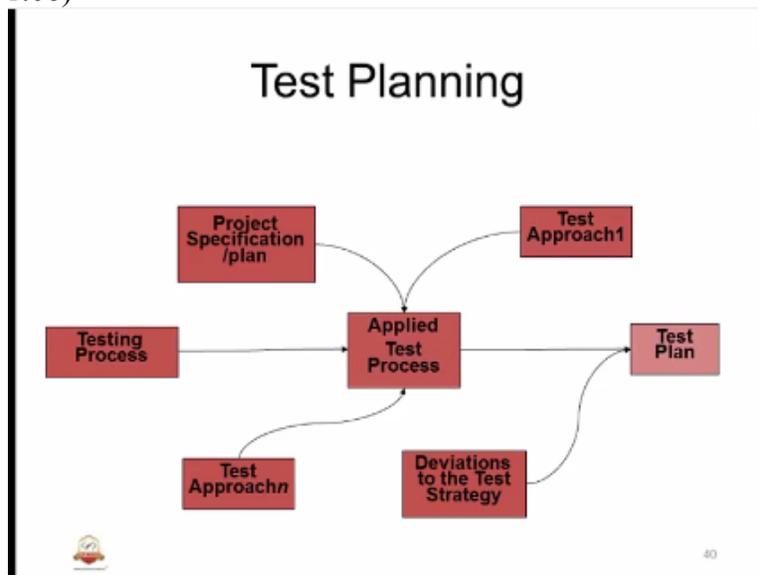


**Session 4 Test Case design and
Procedures
Lecture 4**

Seer Akademi – NPTEL MOU

Welcome to next session of the embedded test case testing, the session 4 will be your continuation of test case design and procedure of the work we have started in the previous session, a quick recap of what we have learn in the previous session, we give the general testing philosophy, analyze requirement what are the steps involved in that and what are the steps that are required for general testing philosophy and what is the meant verification and validation and how it's defect and defined testing we needs a only definition that we have used or V&V recap and what is different between testing and debugging purposes and test planning.
(Refer Slide Time 1:06)



This is the first thing we have to start in any of the embedded testing life cycle or we will start with the specification plan we will write a approach and the multiple of approach the will apply the testing process, so covering all this aspects, then specific to the test plan will applied the test process to write in a test plan, if there are any deviations in that testing related that is called deviation to the test strategy, finally we come up to the test plan.

(Refer Slide Time 1:40)

Test plan's goal we had a, it should be as for this plan and the project plan then purposes of them test plan contents the typical test plan contents as for the (829-1998) though this have a all these elements of testing elements to this, so there also we have definition of a test strategy so or to be, then what are the strategy so it will be address in the test plan in the system level, user level, integration level and component level test and also we had a come to the test plan, so we need to open it again so that we have a idea and a what are the session of we will introduce about the complete test life cycle plan for keeping them, and then a reference may be fixed use of demand and take use of demand all their effort, then verification test plan will again open the test plan and go through.

(Refer Slide Time 3:00)

Basically it has about a taken as six things, it has a introduction about the test plan applicable external and internal references for documents also it as applicable or customer requirement etc.... and it also followed in the test life cycle and verification responsibility, verification methods, activities, environment and the traceability to compliance mapping and to this aspects, we had a compliance additional consideration that also we have to consider the end standards, checklist and guidelines as part of the testing in contents, so this useful.
(Refer Slide Time 4:09)

SOFTWARE VERIFICATION PLAN FOR <XXX>

SW_STP	
2.0	Released
END PRODUCT NAME <XXX>	0/0

2. APPLICABLE AND REFERENCE DOCUMENTS

- a. External applicable documents
- b. Internal applicable documents

|

So introduction we will have a purpose, responsibilities, change control, competence, abbreviations of action, and then reference and applicable documents will have to be highlighted.
(Refer Slide Time 4:22)

SOFTWARE VERIFICATION PLAN FOR <XXX>

SW_STP	
2.0	Released
END PRODUCT NAME <XXX>	7/0

3. VERIFICATION RESPONSIBILITY

- a. Organization
- b. Independence

|

The next section verification responsibility what is the organization structure or who is going to control what aspects of the test or this will be highlighted and off course any industrial embedded testing have to be following some sort of the standard, may I respect the follow which call for independence that's means here independence is a testing team have to be defined by the

developer team, so in that way only when responsibility verification, why this is need is to make sure that the testing is a not aware of the some of the implementation details or the some of the philosophical layout of how it is been done, we should not be buyers, so make it that independency it is always mandatrtrical or independence use for verification.
(Refer Slide Time 5:32)

SOFTWARE VERIFICATION PLAN FOR <XXX>	
SW_STP	
2.0	Released
EVS PRODUCT NAME <000>	003

4. VERIFICATION METHODS

- a. Testing Method
- b. Review Method
- c. Analysis method

Verifications methods, testing methods, review methods and analysis method,
(Refer Slide Time: 05:40)

SOFTWARE VERIFICATION PLAN FOR <XXX>	
SW_STP	
2.0	Released
EVS PRODUCT NAME <000>	003

6. VERIFICATION ACTIVITIES

- a. Planning process verification
- b. Verification of system requirements
- c. Verification of high-level-requirements
- d. Verification of outputs of integration
- e. Verification of Testing results

Verification activities, what are the verification activities, planning process verification, verification of system requirements, verification of high- level- requirements, verification of outputs of integration, verification of overall testing output.
(Refer Slide Time 06:00)

SOFTWARE VERIFICATION PLAN FOR <CCE>	
SW_STP	
2.0	Released
END PRODUCT NAME <000>	10/03

6. VERIFICATION ENVIRONMENT

- a. Testing Environment
- b. Testing tools
- c. Tools qualification

Verification environment, test environment, test tools, test qualification and off course we are going to detail it each aspects of this in the next session or next types.
(Refer Slide Time 06:19)

SOFTWARE VERIFICATION PLAN FOR <CCE>	
SW_STP	
2.0	Released
END PRODUCT NAME <000>	11/03

7. TRACEABILITY TO COMPLIANCE MAPPING

And we need to maintenance traceability of what he is doing to provide in test scoping.
(Refer Slide Time 06:23)

SOFTWARE VERIFICATION PLAN FOR <XEX>	
SW_STP	
2.0	Released
EVS PRODUCT NAME <XXX>	12/03

8. ADDITIONAL CONSIDERATIONS

a. Partitioning, compiler, COTS(COMMERCIAL OFF THE SHELF if any)

|

Additional considerations a pen boning of L patrician used compiler any commercial official tools are used an ender that needs to be corrected.
 (Refer Slide Time 06:36)

SOFTWARE VERIFICATION PLAN FOR <XXX>	
SW_STP	
2.0	Released
EVS PRODUCT NAME <XXX>	12/03

9. STANDARDS, CHECKLISTS AND GUIDELINES

And from the process prospective and guidelines prospective we need to mention the standard checklist and the guideline those all are will be used, during review the same thing need to go through in order to maintaining the review, so after we go through the test plan, we know that there are lot of elements that are part of the test plan, so as part of the first step what we are going to do of the test plan is approved and in place we are going to start with the test cases, test design, test case procedure,

This outcome of test case design procedure is as invert development of test cases for each of the requirements or recruit of the requirement or the specification basically provided to the test case and to map with that test cases we are going to come up with the test procedure, as I said in a earlier session, test cases are theoretical which of group or logical group of the requirements and the specification, test procedure of the practical structure, how it should be done test cases will

tell what to do in specific we will tell how to do it. Test cases are theoretical steps especially for authoritl steps in terms of covering all the test cases.
(Refer Slide Time 08:47)

Test Specification

- Test specification defines *what* to test
- Test specification is part of Testware (testing process)
- Basic building blocks of test specifications are test cases
- Test specification – instruction – script
- Test specification – requirements
- Test specification - reporting

48

So what is test specification, test specification defines what to test, test specification is part of the test ware some of the test things in the testing elements process, basic buildings blocks of test specifications are test cases, test specification will have a instruction how to do the space and all that, test specification will have requirements highlighted, test specification will reporting how much is covered and how much it is space etc..... So today will go through an example of test specification.

(Refer Slide Time 09:34)

Test Case Design

- Analyze Requirements
- Requirements categorization
- Identify test cases
- Identify test conditions
- Apply Test case equivalence criteria
- Grouping the Test Cases
- Special Test cases criteria
- Functional & Non-Functional segregation

performance
timing
functional requirements
...

category2 normal
category2 robust (outside normal, abnormal inputs...)
0, -1, 11..

49

Before that I re write the test case design step, we need to analysis the requirements and requirements could be any of the embedded system product specification of software requirements that needs to analysis that needs to be under scope from this some respective, then we have requirements categorization, here requirement categorized in the sense, we need to

categorize a different part of requirements like it could be timing requirement or it could be performance requirements or it could be many of the functional requirements etc....

So these all are will be categorize on the requirements prospective, why we need to categorize is that, these will move out the test of job in terms of this understanding, suppose because why I coming I suppose we have a team of four suppose, we can divide the team instructor in such that one person concentrate on the timing aspects one person can concentrate the functional requirements, one person can look in the performance requirements, one person can look in the timing any other memory related or anything, again we can divide that in terms of categorization and abbreviation it is a good practice that they have a requirement categorization.

Once we have the requirements to categorize after understanding the or analyzing the requirement we are going to find it with the test practice, so as I said earlier test cases will have a test identifier and the test conditions that is for identify we have the different condition how that test cases will be used and then we have test verification, then test inputs are nothing but to find the test value that can be spread to the discard for that particular requirements then we will test output and what is the test output instructed of the particular test case, for doing that there are test conditions which will be applied for identify test conditions that is the next topic , next one is the test case equivalent criteria once we have this set of activity done for a group of requirement or specification has to write, these are the group for writing the test cases, that means we have an understanding of how test case should be then so we have started with the test cases design, once we have that then we are going to applying the criteria here criteria means.

I will explain that in a particular how it is getting done, suppose one requirements if suppose take few value of the requirements says that it can take 1 to 10 that's the explain, 1 to 10 inputs, how I going to test it? So definitely the system is suppose to work taking the value 1 taking the value 10 taking value anything between 1 top 10, so here have to design in such way that all these value are the exit, it does' mean that all the value in incremental manner should be excide but the test case design should to the criteria specify that it will excide enough of the system to pass through all this, that means if I have suppose 1 to 10 then I will do equivalent class in terms of boundaries, here boundaries is again know the requirements let us take 1 and the requirements can take 10 to lowest is 1 and the highest is 10 and the moderate value also will find some integrate value, that is called the equivalence we are going to do, so we can do a equivalence of a 2 and drive to the 5 so will select the values as 1, 5,10 this will be equivalence criteria or normal range of test data, then coming to the, there is one category the next category of a Test cases design in terms of so test casing is a robust that means it could be an out of outside the normal or abnormal inputs etc. for example they requirement says that a system can take the value of one to ten so we know that the system should work for one to ten also we should make sure that the system will not work or the system is not going to faces the values any outside one to ten so what we need for the inputs for this conditions it should be outside or the abnormal diagram the input values could be 0 or the input values could be -1 input values could be 11 etc.

So these are something like outside the normal or the abnormal input or the robust input there are chance that the system may not accept you need to understand this why because suppose a node relected node for a fan will have 3 to 10 as a preset value so how do I feed 0 how do I feed -1

how do I feed 11 is the questions again I will explain that this again depends on the test horns how we are going to develop it I need to figure out a way of doing that or providing the input. If that is not possible then we need to justify using different cases here there could be manual go through of that or I am doing simulated input we are particular because software is responding for typically we are useful etc. so in that way I am going to designing so all this will be part of the test cases design what are the test cases what are the inputs what do we conditions and what is the expected values for hint of this category, categories test okay.

So now we know that we have done the requirement analysis test cases identified test conditions all we know test inputs we have define all this will be we have done next once we have done various test cases for example for the example we will say 1 to 10 it similarly requirement to will have a different sort of a functionality that mean we have a test cases of may be 20, 50, 100 it depends of the particular requirement what I am going to do now grouping.

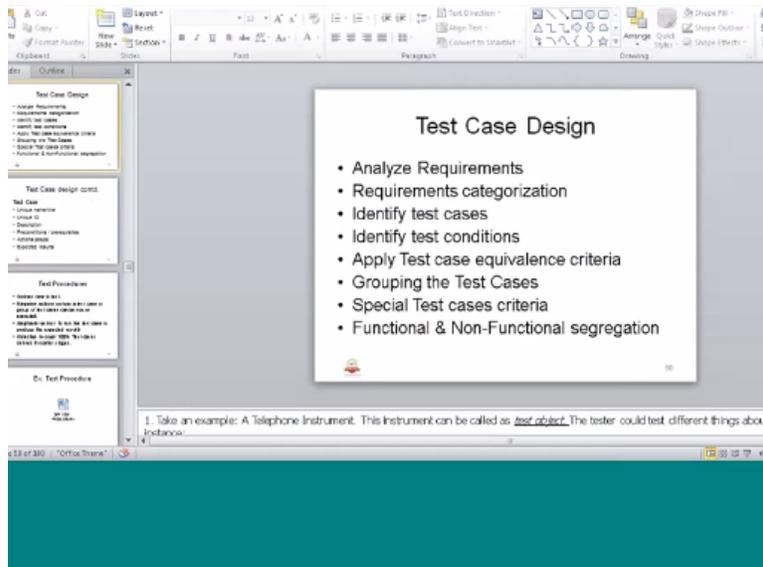
So grouping is again based on the functionality and perform etc. it did not be purely based on the requirements it will also be based on the executions criteria suppose I am going execute one set of requirement and the next set of requirement is also could go to the sane executions path I can group it so there a chance that I can group multiple data of test cases for multiple requirement in this grouping, grouping is very important thing we have to group the test cases.

It is meaning full and it is particularly and report based the easier particular entire will be in group test cases of course some time what will happen is grouping, grouping the test cases may be remember so what we have to do is we have to consider some of the test cases are special categories we may have to un group it or we may have to categories as an independent test cases it does not be fall as an independent it need to be fall as grouped cases.

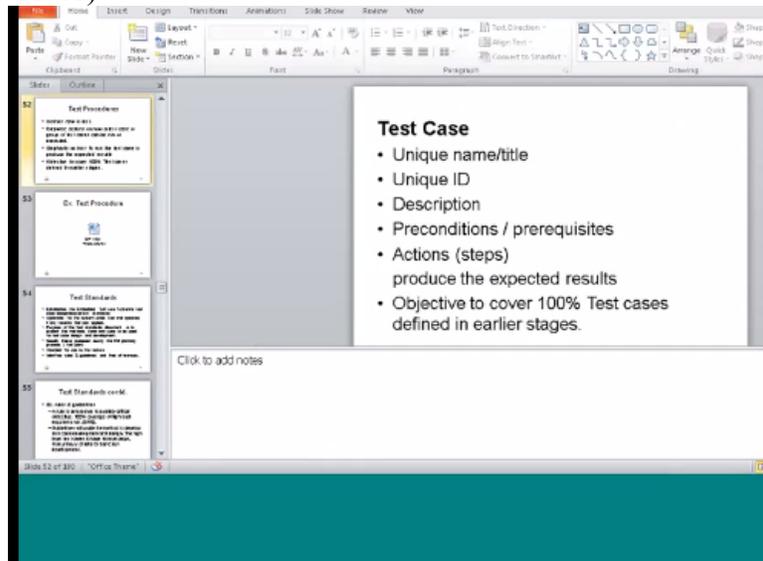
It can be a special category test cases once we have all this then we are going to have a basis functions of the entire test in terms of functions & non functions segregations we know what is functional all the individuality given or product given requirements of all functions, functional and nonfunctional requirements for performance related or timing relayed or what is the group and how much it can tolerate in terms of over voltage or it could be size of the memory etc.

So this are some of the non-functionality requirement all this will be part of the complete test cases based on the basic is to be done this is a thing that has to be done with the help of the team as said there is going to be test organizations who will have explain in a test plan I think I told that which will have the structure defined.

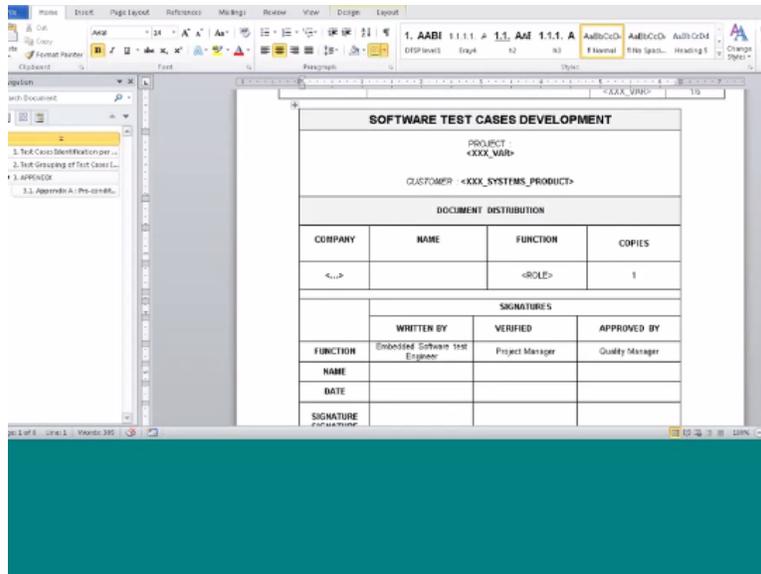
(Refer Slide Time: 21:18)



For test cases development after that we are going to develop the test case design will consider an example of test cases design I think I will repeat this page probably after going to the test aspects that is in the test cases.
(Refer Slide Time: 21:41)

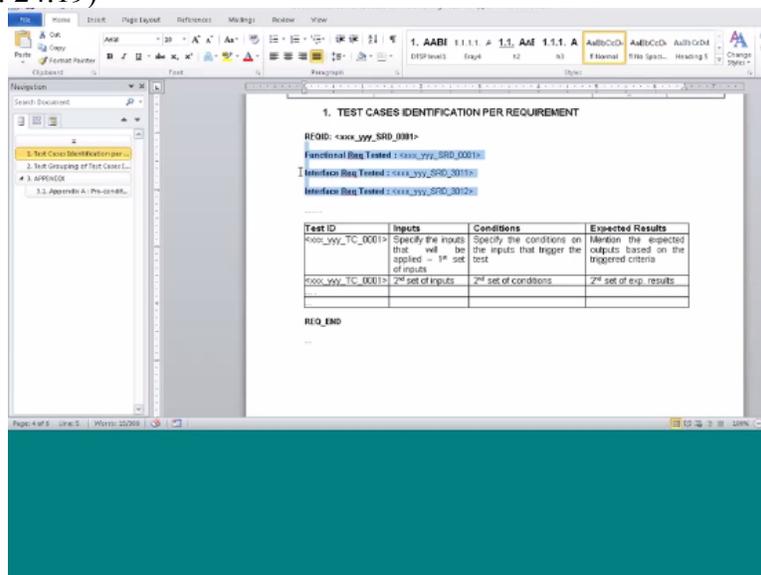


Test cases again okay in that example we are going to see a unique name type logical space in the unique identifier for each of the test cases descriptions preconditions or the prerequisites that is to be required for the test cases preconditions is some like which has to mandatory followed for doing the test that particular test that it is a predefine the conditions for that test then we will have the steps which are test cases steps then we will have the expected result for this particular test cases after giving the specific file after it will have an understanding of test cases.
(Refer Slide Time: 23:01)



Now we will go through some test cases examples I hope I able to check software test cases development I will name it as STCD which will have a least version of 1.0 and that product name which variant between the embedded system and on the variant then it will be named then we will have the calculate as you can see on this slide for the test cases development and we have the DOCL at the end of this there is high light portions which will high light the preconditions, precondition are looking like the conditions that must be available for executing that particular.

(Refer Slide Time: 24:19)



How test cases are identify that is of the requirement whatever I am showing is an example of typical test cases adopt this will not busy all the time but mostly it cover the aspect of embedded system test cases is an example so we will do first we will test the requirement with it is ID what are SRD each of the requirement will align then in and what is the functionality if the requirement that we have to test in this particular test.

Then any interface requirements we are going to test here interfaces requirements basically a software requirement suppose I am going to give an example of how SRD is be like basically it will have the couple of part one part will be highlighting the basic operations functional development the other part will be surrounding this operations development surrounding means it could be an input to this requirement or it could be a output for this requirement or linking requirement or an interface requirement interface could be interface.

The primary requirement suppose the requirement say that some based on some value some action will be taken places so this value how it will going to come till what it is going to be attach the black interface requirement so interface requirement will specific about the interfaces for a particular set of requirement functionality will high light the function of the predications all this will be listed at the part of the test cases identifications because this test cases identifier.

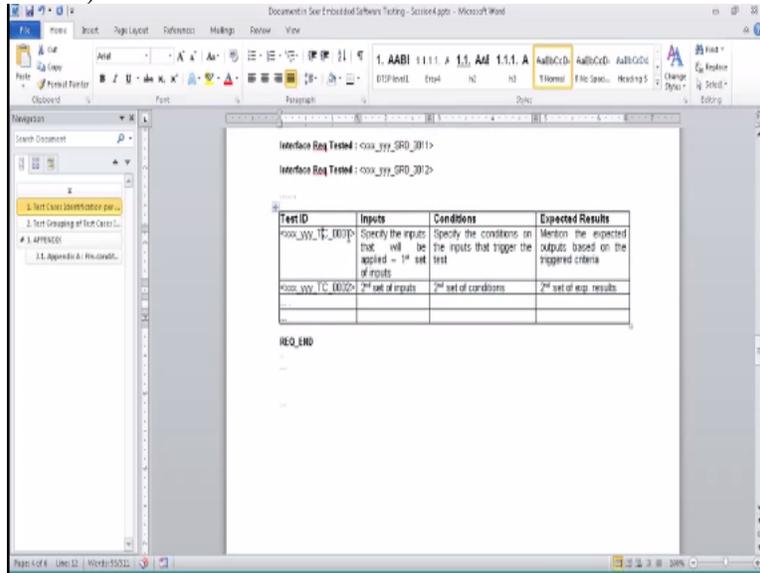
Whatever it is using the table we will cover up all this requirement all this testified and test cases various format of test cases design it has a test ID it has a input it has conditions and as I said each test cases will identify the unique ID or it could be any TC 001 whatever it is then for that particular test what is the input has to feed to execute that test then once we provide the input first one you see that the first set of test here I specific the inputs.

That will be used for that apply then the conditions once we provide the input what I have to do should I switch on something etc. so the thing will be specified here the conditions that or aligned with this inputs will be mentioning basically it trigger the test at the end of the result or at the end of the test based on this conditions so what is output that can be specified that is highlighted here and then expected result to mentioned the expected output based on the trigger.

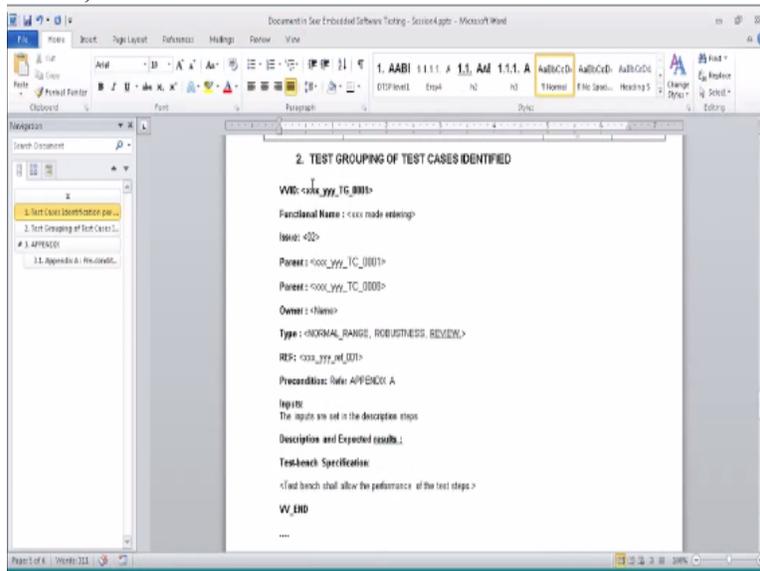
So likewise we are going to have different test cases it could be one, two ten as said for the example requirement or the requirement could be functional requirement identify all the functionality particular requirements and for each of the requirement or any of the requirement it could have an interface requirement as I said interface requirement is surrounding the functional requirement but it is specifying the interfaces.

That are required to satisfy the primary requirement the interfaces could be any input value or any output values it required so all this have to be covered in terms of addressing the functionality for the function required so the test cases primary some like there are four column which will identify test case have you see the product name followed by it is variant then have a TC team then again test cases so we can that sold be all comes under this set of functional requirement and with specific test case has input so what are the execute or to specific the inputs this could be first set of inputs but this test case on the condition that input regarding the test cases the conditions that I need to do it so really accept the first set of value for this conditions, for execute what is the output the output could be the value or any non-ring output or anything it could be so all this happening in to the last column. This will complete the one test case. Similarly will have test case 2, test case 3, etc. test case 2 will have a second set option it will have its beyond second set of completion it will have second set of execution value. Test case if it is 1, each case will be by itself will be all input we have that will have expected results after that conditions have been executed so all these steps to be formatted for this case. For this requirement this test cases to be listed on. So this requirement what have highlighted here,

nothing but one requirement. Likewise it will have multiple requirements. So likewise I am going to different test cases so this requirement we end of it all the test cases identified in the perception. So this test case identification for requirement for the each requirement will have identified the multiple test cases.
 (Refer Slide Time: 31:48)



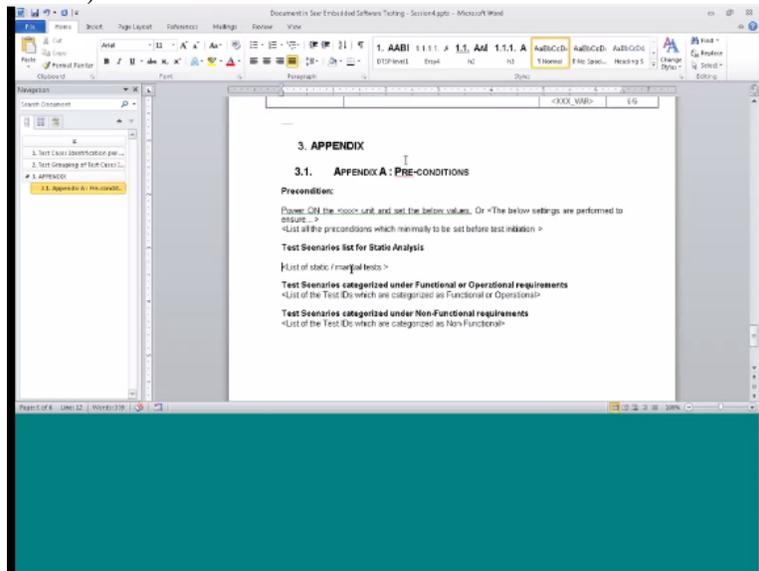
Okay, now we know that there are number of test cases if we this is 001,002, till 20, 30 whatever it is, and I have identified as I said in my earlier slide, we need to group the test cases. So grouping is the second test case of next section.
 (Refer Slide Time: 32:11)



So this grouping will have its own identification we can tell that the functionality, as I said the grouping is really based on the group of requirement for the functionality, and some criteria which will help particular test cases, carry on this test. That is why we made it as functional name of that group. So, I put an example, more entering this is one group issue is first issue; second issue and you can have it or doing. Depends on the configuration and it may okay, in this

group what are they parents going to take care. So I have requirement test case idea one test case 6, so these 2 parents I am going to cover. Parents are here is nothing but pointing to the test case. So this test is for group for basically, then I am going to mentioned you owner, who is the responsible for doing this grouping, then we have what sort of a grouping it is? It is what sort of test cases I am going to cover in this group. The result it could be normal range it could be robustness it could be review or it could be manual for everything. We can mention the particular type which is applied for the test case sorry this group. One we have a reference, the reference could be anything in the requirement, it many of the just write on the test case and test recruiting, to we need to probably refer some of the other document like circumstance of any of the system level in for mentioned. That is for understanding purpose that we can know in particular. The next session is sorry next highlighter is the pre conditional. As I said pre conditional it has to be for particular case.

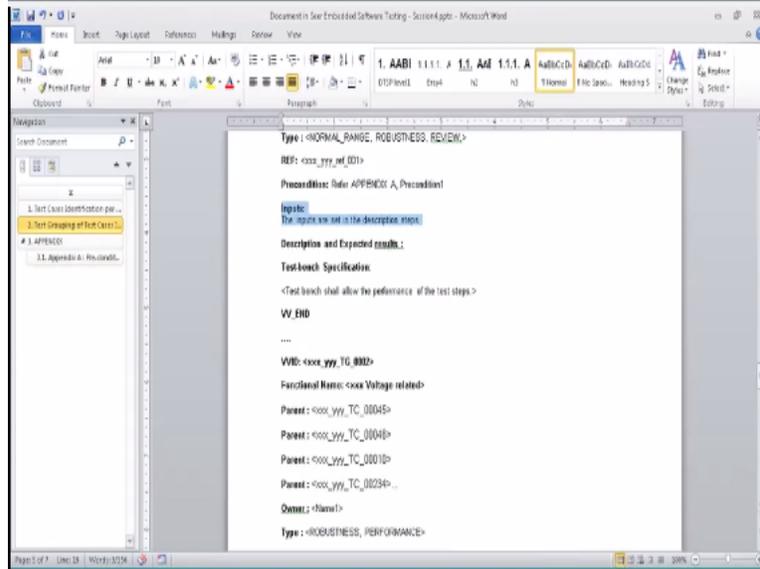
(Refer Slide Time: 34:40)



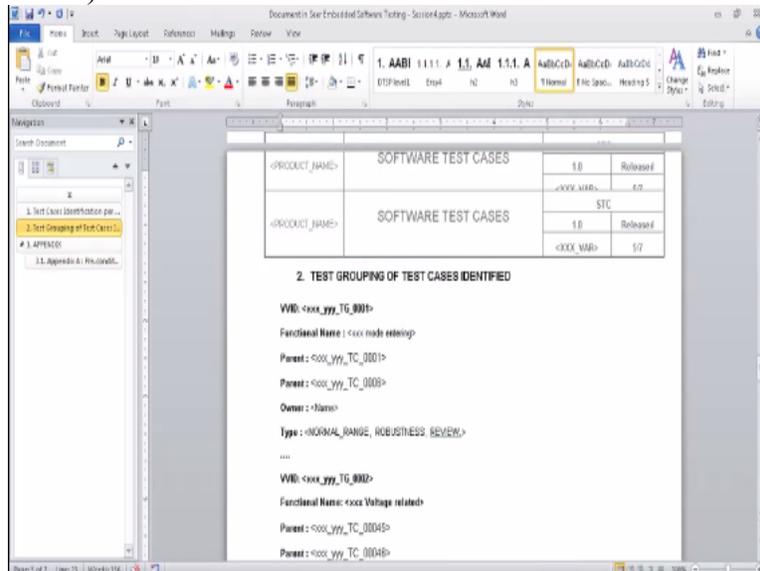
We will just for example appendix a pre-condition is 1. We can have multiple pre condition. Here we have instead of 2, 3 conditions so I am going to use first three conditions. That is per condition one is used for developing this test group or grouping different test cases, test case 1 and 6 and this pre conditions have to be implies before or by taking of the test case. What is the pre-condition? Pre-conditions it could we need to power on the embedded developing and one of the value or velocity carefully performed some of the default condition like instrumental value turning in to some or else times to becomes value or some of the discrete should be enable or the power has to be 3 volt 5 volt all this will be part of the pre conditional.

This is all you can use which minimally to be before essential. And that is about pre condition and we will come to the section. Similarly I am going to have next group in terms of let me say test grouping of test cases identified will identify the next test group I am just going to write down all this for that this something like development with understand. For this functional of the whole thing the next one will be the test cases. What are the test case that is defined and then for and to have the test case that is identified. And that is name of the owners. Here the particular

thing one I was mentioned by normal range, robustness, and review this them and then to cover the test cases. This have to be done consider as the basically there will be the tolerance of constant table, or the user of the requirement that will be use. Next one is the pre-condition is nothing but the decimal set of values are default condition that use to be recreation have that is apply the voltage for at least so what means while doing this before start of the test I must make sure that with voltage is 0. First I check. That is what the pre-condition do for. (Refer Slide time: 40:22)

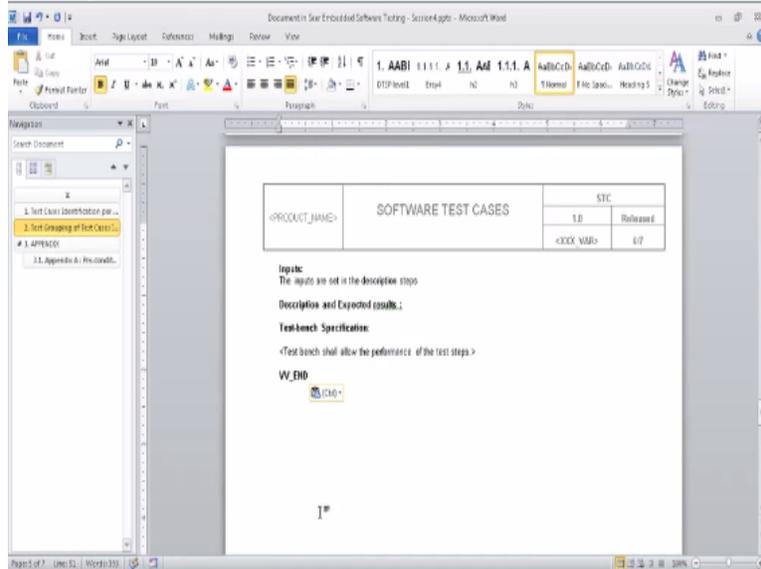


So that is about the pre-condition okay, so next one is inputs, the inputs for the test groups we know that what is the status, what are the test cases? So we need to highlight that those inputs that can be sets that means combination of the entire test in terms of this test will be highlighted. But one highlighted the input and going to describe and I know to provide. (Refer Slide Time: 41:00)

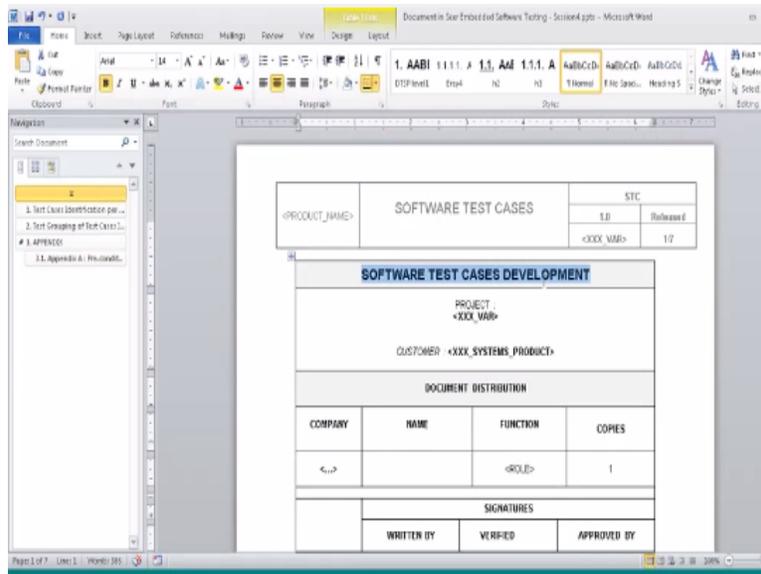


So basically this will have again that test cases what we identified this but in the grouping what I have done is I have highlighted multiple requirement with the flow of expected results. For that I

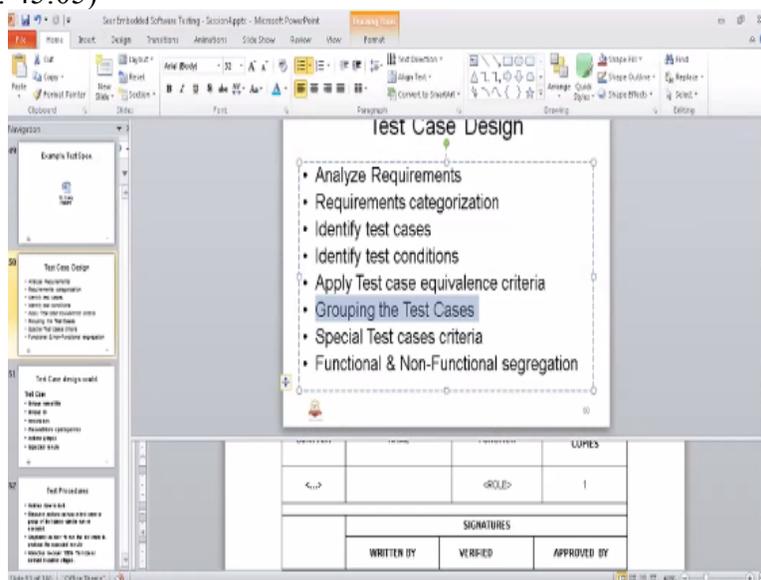
can continue from the test case from 1 to 10 or 20 whatever grouping that I have done. So that is the idea description and expected results. Identifying the format of inputs and describing the conditions. So before end of this test group, there is the test bends that is to be highlight I will later tell you what is test bends? Basically the test bends are set of the time going to exist to this set. Test bends could be automatic test, the manual test all that will be in terms of test case. Same thing we will to have it for all the group of the pre-condition we will have.
 (Refer Slide Time: 42:23)



So likewise it is going to continue for all the grouping of the complete requirement. So that is about the requirement the first instruction we will identify all the test cases for each all the requirement so we have identified the all test cases from all the grouping has done each of all test cases will be covered in all group. So the test case development is appendix.
 So the end we have appendix either you can use any pointed to any of the document in this or you can use appendix of highlighting the per-condition. Also we have the dimension also the appendix the test scenario is static analysis. As is said the test bends that I am going to use could be a code. So it could be any static analysis so I am going to list out what are the test exactly the taken care of the static analysis so what are the test each of the categories under operational of functional requirements. So what are the test factors groups each have taken as the nonfunctional requirements? So this will power by a list of complete set of test. So this is something like that coverage metrics appendix. Cover is under what? The separate system here I have power and all this is up to you how you going to design. Test matrix is test sorry test coverage matrix or whatever you asked. So it could highlight the entire test and the grouping how it is done and I am going to list it out. This will be a complete test case development or test case designed architecture. Please ask any question you want to have if you can ask each of this instruction. This is an example of course I am going to provide you next exercise.
 (Refer Slide Time: 45:01)



So you can exercise this if you have a clear idea that is about an example of test case to be done. (Refer Slide Time: 45:05)



So all this part of the test case design. I have put some example something like I am going to explain. We have gone through a test specification example if all the aspects you can have test case design. So one example we had considered to be telephone instruments, the instrument which we want to test it. It designs under need the software something called as test object. The tester could test different things about instruments for example telephone should allow user to talk to another phone. Telephone will have the features of redial, flash, conference call, or and another call so have considered available or not. If it is available and they will use as per the document whether they have documented earlier to aspects so one thing use that the products of such as it complete for only when it is appropriate the developed. We have developed means the product should have a complete user affected. User acceptance it could highlight the manual the document how should be used that is the artificial part. Along with what it is possible of doing. It is an operation or it use. Similarly how we are going to use

that instrument, how long it convert how much data instrument is store or is the number there is will be part of the requirement analysis. So I would say understanding of this plan. You have this above items will play the tester should have an information about what it is expected from this instrument that thing instrument has to be understood by the test plan. So these prospects of to be continued in the system of the example, that is the tester should have an understanding of the product, what are the test case like what it is designed, so we cannot apply on the test case.

Okay so we have an example I will have an unique name, unique ID, descriptions, and preconditions actions, that means the steps and the expected results which is the last column so it is what required to finish the test case as per below that we need to do specification of the code, all tester should follow identical and the ID must be permanent, regarding any instruments will be added, so these are the important part.

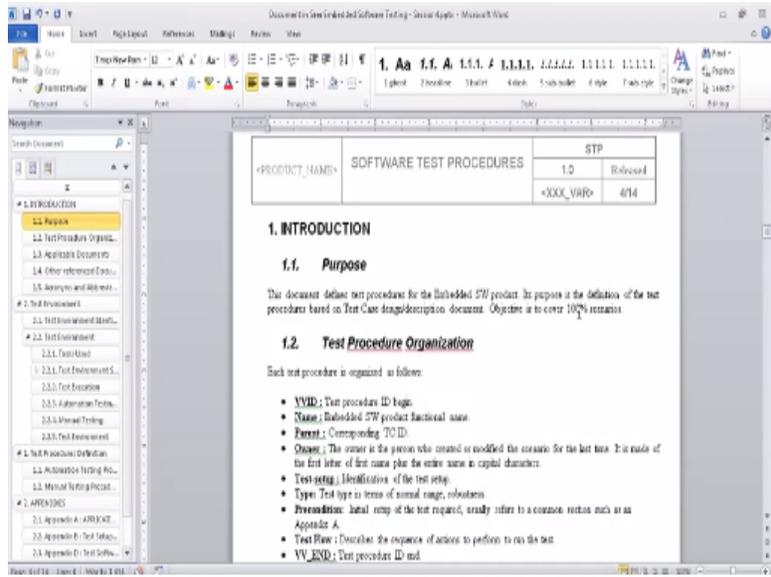
So suppose I have say this case as 56 or the last column of course so we can collide, we thought that what we will do is we are going to integrate it, by doing that so, still we should not as per the number have this as a completely, the next is the accept the one by one, first we should have a multiple tester, who is depends on which procedures and all that.

So it is very much inculcate in change, so this number the definition is if you want the this word in the lead after any number, suppose you want to ensure the certain number, I am going to do the configuration of the last two numbers, so we are going to do the number which should not alter functionality that preconditions that will take all the conditions as per the code of the test case design.

So this is all about the test case design, next we will go to test procedures, we know that we have developed, if we done what should after the requirement, because all the aspects from the theory is about how test case design can be expected, to achieve expected design for the particular test cases we need to have a practicality of that given test condition or the difference that are used etc. all that will be part of the test procedures so it defines how to test stepwise actions on how a test case or group of test cases can be run or executed so basically emphasis on how to run the test case to produce the expected result objective to cover 100 percentage test cases.

Defined in earlier stages we know that for test case it need to workable so test procedure has to be 100 percentage to cover the test cases because we have developed the number of test cases all this test case have to be then or the other way should be executed how we are going execute this all test cases defined in earlier stages test cases are defined as the tell the user or the test procedure how he going to test on the test so we will go throw the test procedure

(Refer Slide Time: 53:47)

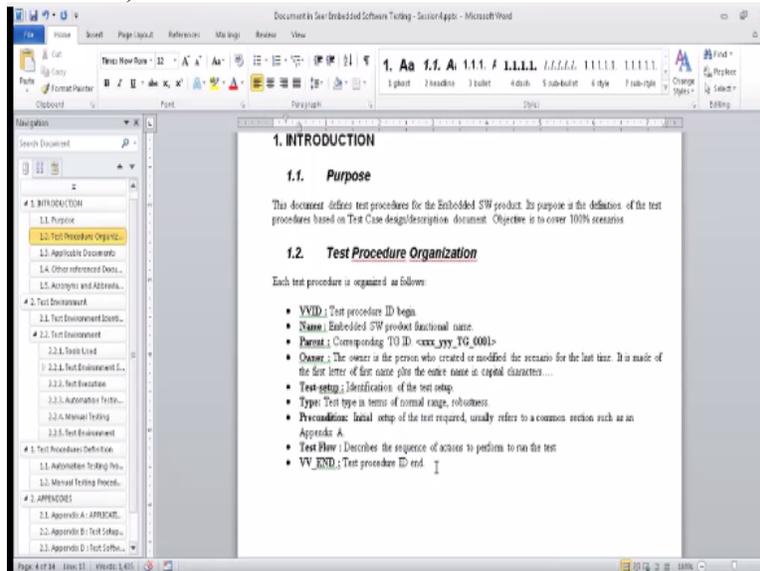


This is the test procedures are done it by using the test cases he understand by is there an design then test cases the test cases are something like design, test procedure are something like making the design work obviously the design work obviously to making the design work as to be made more to in tester so you can place header all that aspects of standard so test cases can be fined and test cases can be test plan.

We will go throw test introduction, test environment and procedures definition and we will have introduction we I well explain about the test procedure organization and purpose that design test procedure for the embedded software product to propos is the cellulites that use procedure based on test cases diagram to produce 100 percentage to recover are for the test procedure it must be recover 100 percentage all the test propose are used VV_END test procedure ID code or the ending of with a VVID and it is up to the organization of the user how he is going to write it in general it is about the all the test plans so VVID about the test procedure ID combining of the positions in which teat case that we are going to cover so one issue is there it will be the test board we know that so test cases is a procedure we have terms about the test plans so example test group one the each of the lining has to be covered by use the test groups the person who creates the test plans it is made of test lecture all of the test plans in terms of each cases whatever you want and if independently have to enter the test of the test cases it must be don independently with no hiding so the in this critical as we have seen an even the test procedure are right there or the test revues are totally independent basically the one who developed of the test sceneries well not be independent it will be executing other persons for the test cases so it will do it all and we may understanding the aspects so that why to have independent during all this thing carefully the actives.

The one is the test setup as I said before what I am going to do is we will identify next is the test file and nest is developments of test you may ask why I have repeated all this , it is because every time who will done the execution who will develop the scripts well not have to go throw the test case or the script he has a clear idea of the test cases so he will use the grammar of teats

so that only he knows all this inputs in this cases and as same part of the group of the testing when as the other
(Refer Slide Time: 01:00:00)

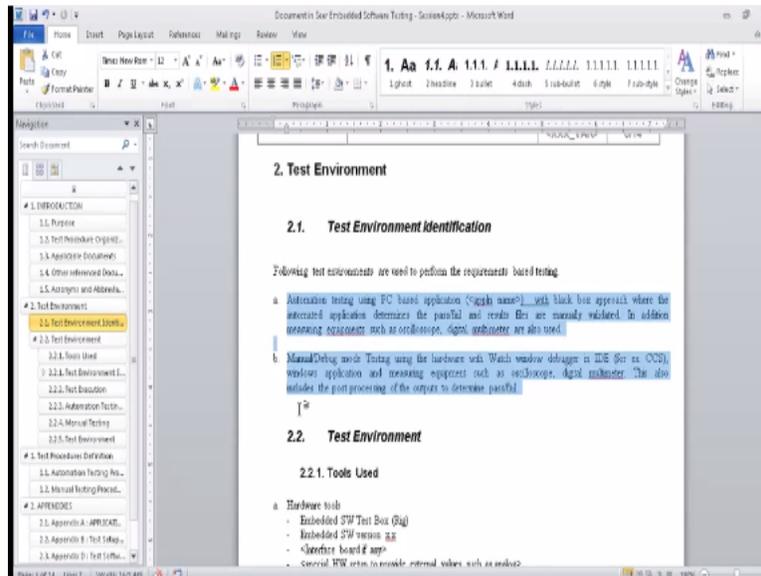


One of the other important fields of the pre-conditions can be operated or degrade as I said in the test case development as given it might not have complete practical knowledge or will be like setup that is going to be use or the overview of setup that is going to be use whether we need to be use that is what we need called execution of the particular setup once all this setup is done we are going to have the test flow, we must need to do the practical test, the sequence or the action that needs to be performed on the target or on the system to execute the particular test case in group, so this is how test case works likewise we will have multiple cases.

So the next one is the applicable documents which are the software verification, test case document and customer supplied document and standard checklist all these we are going to use as a basic inputs in the applicable documents and any other referenced documents like IEEE.

So next we have all the abbreviations, and now especially these three sections are very important that is why this document as to be separate in several places the application as to be there not be any separate document, while doing the develop testing and the same document they are going to execute and provided and mean while we are going to control.

(Refer Slide Time: 01:05:08)



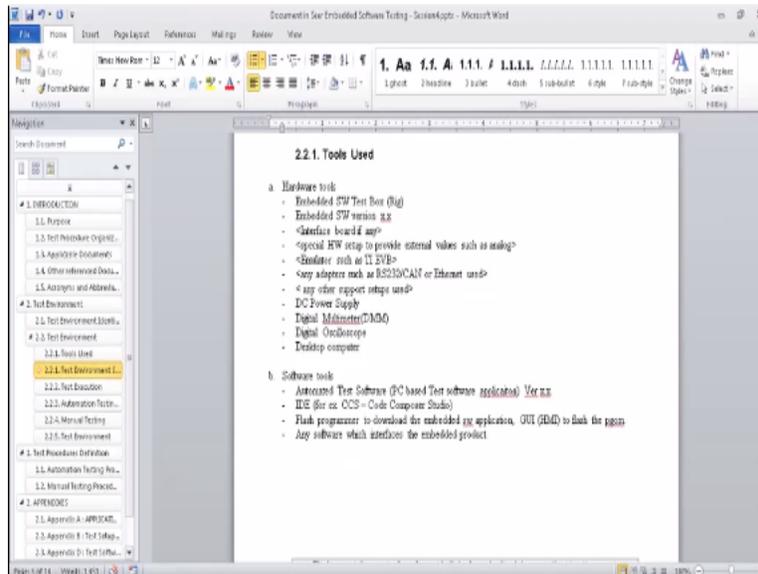
So in many cases or in many domain this is mandatory that we will have a separate design in that again these three section are very important those have to be added to the information, so what is the environment for doing this procedure is very important, test environment will have a normal timing, and it is designed to test, how it can be used or how it can be identified, in environment basically will have two types one is automatic system and tool used.

So you know the automatic user is human resources we will use that, we will have document for this where we will have scripts running all the time and we will monitor output in the target is done, we could have the manual mode which will have the any hardware content.

So how could we find the target for the environment, so this tool can be used, basically we categories it at manual testing, as the part of environment we have tool what are the hardware tool, so I think in next session I will explain this about test and particular target how it is going to be used in the environment and the abbreviation are monitoring etc., so we need to identify the environment,. The environment will have tools the tools could be have a software tools and the hardware tools, the hardware tools have the target code and the programmed embedded software analog and special hardware can be there and it could be analog DC power.

So we have digital multi meter, I think I am going to have apply it for embedded term which are used in the function, and I am going to add all these one by one, so that you can just have a recap on all this, those things have to be in mind for an embedded system software, it could be an analog or special software and for software tools we will use target connected to some tools, any software which interface the entire products , you say the monitoring the internet, or anything, so these are the things involved in it.

(Refer Slide Time: 01:08:39)

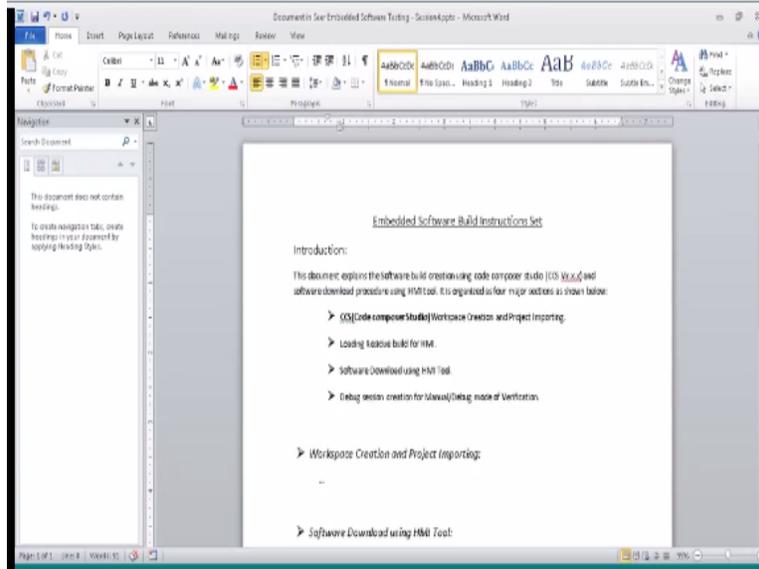


So, next once we have, the software and hardware tools, how we are going to have made the setup, the setup will have a connection, setup will have an diode probably we can see this example in next session, similarly on the host we can have the target based setup, here you have highlighted the set of host connecting the USB into the host with the adapter is the example basically and etc., so the other important task of the embedded software is you may have to build as the aspects purpose so we need to have a build in instruction available in the software, and I will tell you what is software build.

Basically entire software program is going to decide the embedded target, to use you will have the target file, and which will be programmed eventually on the target file, so to program it eventually is called as image an image is nothing but an raw binary file of different format that are used.

So, we need something like USB, as is said this will be the ideal which will be useful in programming or developing the code, compiling and it will be executed then we will load the executed using XML, and it will use the program to connect to the system and program the image.

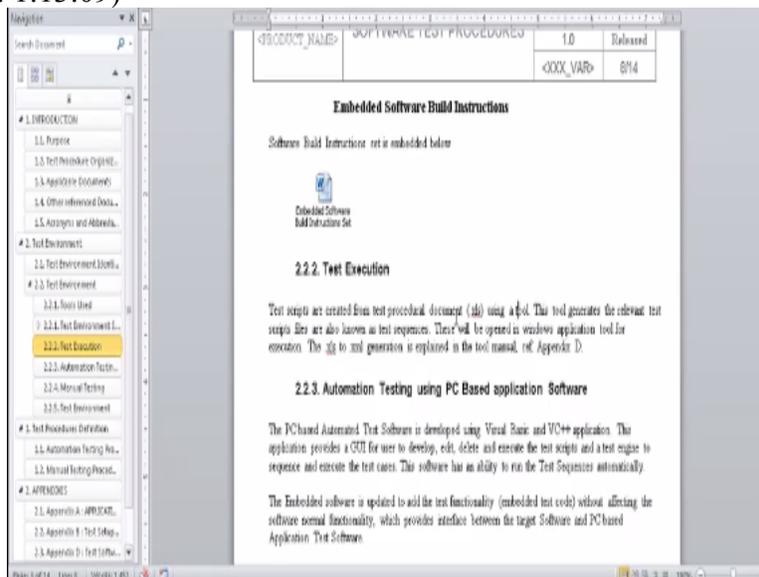
(Refer Slide Time: 1:11:46)



So then we will download that, then we will have a debug creation or debugging, sometimes we may have an additional cards which is imported, then how you can download the additional card, so these are some of the steps involved in the data core, and this will be the part of the test procedure section where the section are highlighted.

So next will be test execution, how am I going to execute the test, so for this could be used as the any of the scripts are will be posted much, so these are created from the test documents using the tool, and this is the example I can have it, basically the test procedure the steps are used and one example is that all those steps will be mentioned in the excel sheet.

(Refer Slide Time: 1:13:09)



So they will the highlighted PDF of test execution so I will again tell all the scripts, and that excel sheet will be used as an documents sometimes with certain columns with highlighted values, and certain columns highlighting those inputs and outputs and certain additional columns having the expected values and the scripts will have and generate the script and also the figuring values and will execute, basically at the inputs from the test case documents can be in excel sheet

and because of the values are in the sheets and it produces the results and it compares the results as per the expected results as per the excel sheet and it is generated.

So the tools generate the styles that is also known as the part of order then this will be open in the window for execute it, so it may not be in this way it's just an example the written methods are used. The XML is the next generation or nay application whatever it is, so the next one is automation testing using pc based application software. This is a basically of application which will be used for developing the post base tool, so I put the example here typically how it is done, this is where automated test is developed using either 50 percentage of application , with the application software provide GUI for user to develop, delete and execute the test scripts.

As I said the test scripts are nothing but python or according to the script, these script can be together or taking care of test scripts and it can execute at the pc of the application and testing in the sequence or execute the test cases which is been part of the pc based application. This software can execute and run in the arithmetic, that means we can higher badge to execute the multiple application etc., this is what the testing will be automated.

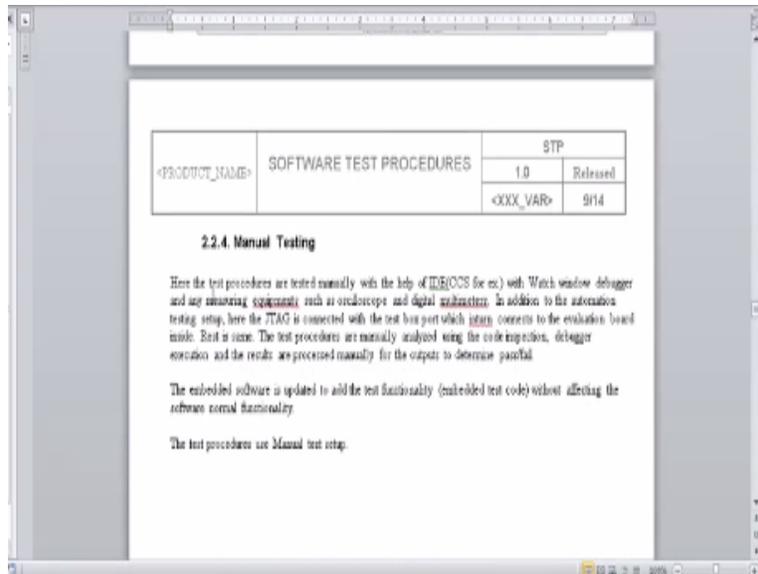
The embedded software is accelerated to add the test functionality by without affecting the software development functionality, which falls in the interface between the target software and pc based application software. So here what it means is, we have developed the pc based application and this should be milting for this guy to connect with the targeted, so this connection could be over anything, it could be cant base or it could be RS case any of the application we can use it.

So basically the interface as to be there, for that the pc the application based in the pc can be used, in order to communicate with the targeted. So in order to communicate with the pc there should be equal or corresponding is responder or the reactive updates at the target system, so there as to be something that is called HS code or test hook. Basically it will have bits of code part of the embedded software, which will interact with the pc based development. So that is what it is been told here.

The new software is updated that is basic core system within the embedded target, it will have additional test functionality it is mostly called as test core, so without effecting in these software normal functionality it have to be test point in the terms of the test code which will block interact by result, this quality is there in the embedded software.

So which part the interface between the target software and the pc, you may take up and an example of the based state or you may again reveal it, because these are the important aspect of embed software testing. So we will read one more example, there could be several test plans where those systems may not be the test software those system by itself have been designed to support that target connecting with the update team, this is the new for test software,

(Refer Slide Time: 1:19:35)

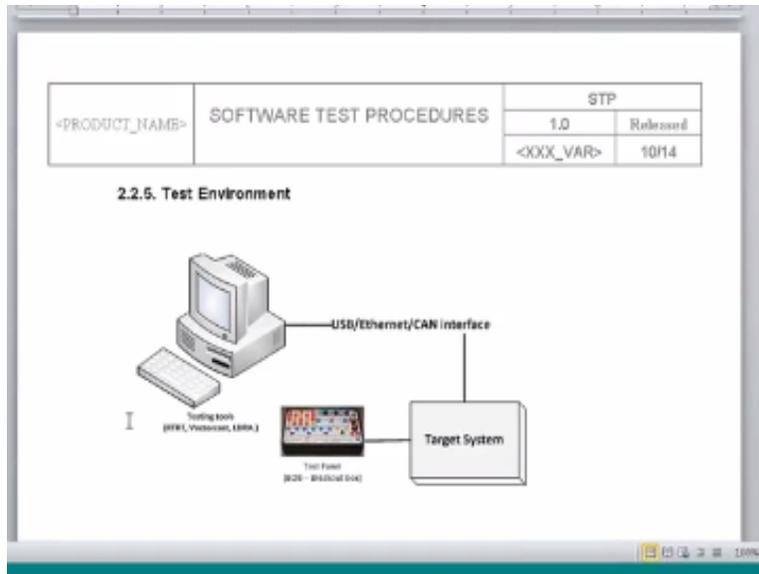


The next one is a manual testing, here the test procedures are tested manually, previously we have seen the automated test is based on the pc application, the next group of testing we use it manual term are. Manual testing the target loaded testes manually, so help of a ID integrated development the normality test code compares to the all life cycle, these are some of the ID used that are used respected to their costar.

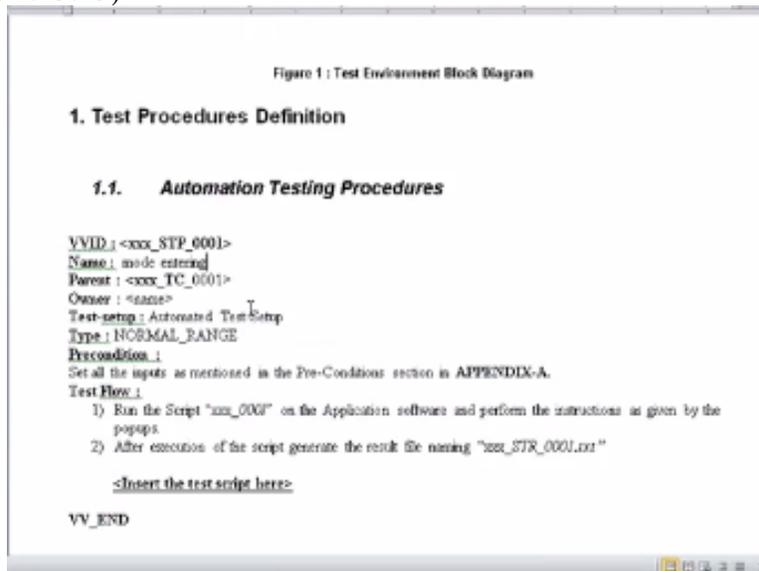
So with the help of debugger and any measuring equal can set assails code as simultaneous, we will have in manual here testing is done. In addition to the automation test attempt, here the data is connected; we know that in automated testing it will debug and access the memory, these are the target of the black box. Here we will use it as a white box to enter the ID, that id is a remittal, the end target is using a J tag, so the j tag is connected with the test box which internally connects to the evaluation priority or the target core.

So as far the end to end instructor connect to the process here, test case are manually analyzed using the code instruction for the debugger execution or we can also so an a automation on the debugger, the id suppose, I mean there are id comment with the help of them it is automation. We can detect that execution of the debugger in term and the results are possess manually, that means the result are carefully with the help of logger of the id, the outputs are determined while in the aspect. The embedded software is a fixed here also, like if the function test core integer here also,

(Refer Slide Time: 1:22:15)

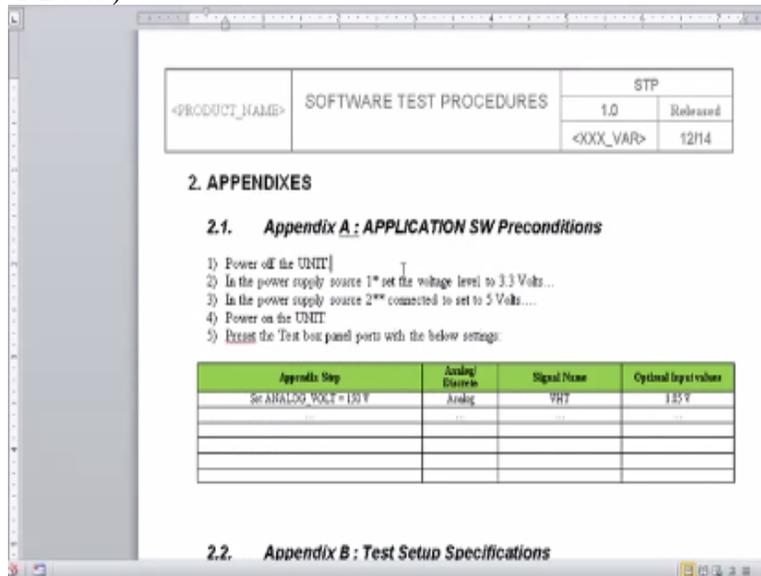


We can restore whatever test cases are used, which will be used in the manual test. Next one is a picture with the thing test scenario, you can see here system environment, this desk top or pc will have testing tool id or in automation tool based on the application that has been pc, and that have a connection in target, target load can be left time in the type of the test panel or the break of box, so in this the test environment will be used.
(Refer Slide Time: 1:23:23)



So here the few examples I would put, the specter is same basically the automation test procedure and the manual test procedures, this set of will use the automatic test setup, whereas in this case it will use the manual test setup. We know that in manual test setup we use the Id or data based and all that, in automated test setup we will use a black box connectivity testes and rest is all same, the tools can interrupt the script here what we done here, so that we complete with the test procedure in early we got manual testing, we can also have the quick based on the manual testing. So that is little bit with this possibility of which it related, and we use some of the environment specific preconditions.

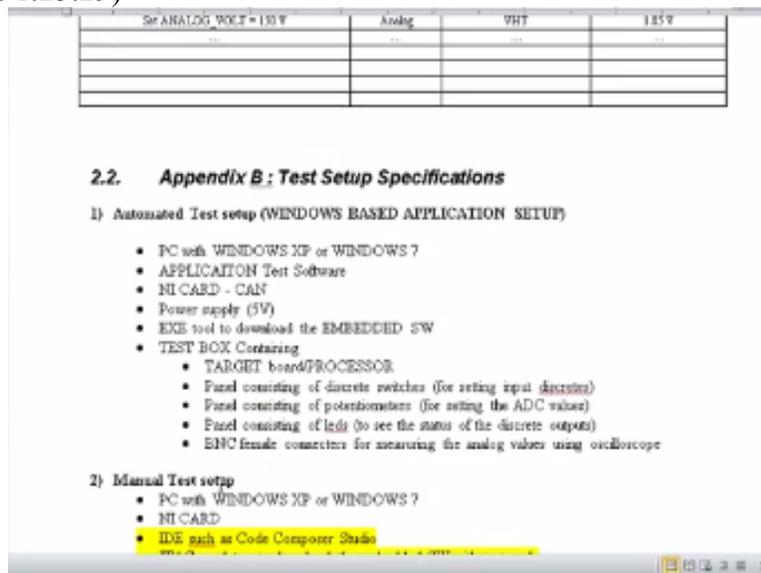
(Refer Slide Time: 1:24:49)



As I said a specific precondition is with the specific values will be highlighted in the test specification, those could be for example, the different voltages setting up the different voltages and some of the analogs setting up in the list, so all these will be part of the preconditions that are resulted specific test setup, specific type of test.

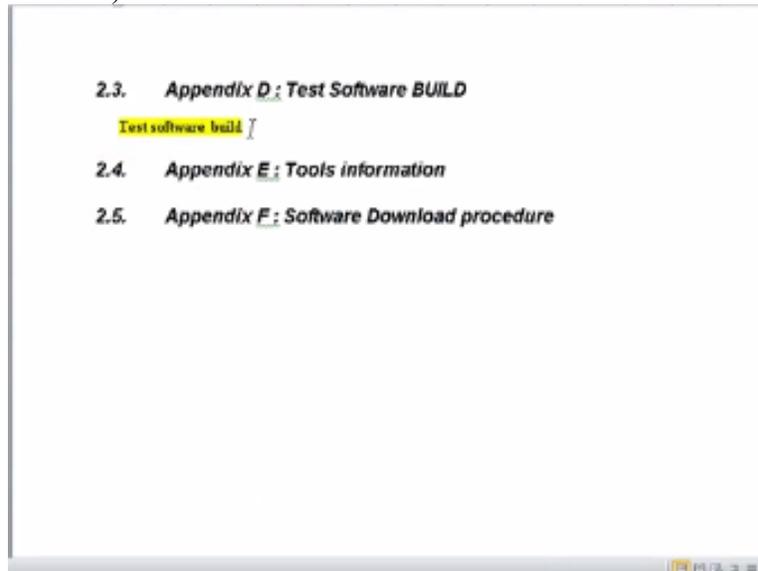
Here we have put the automatic test setup, what are the constraints these details are application software it may make of the CAN, NI CARD communication, power supply and execution tool is founded in the direct software. And test box, evaluate board will have a target board that aspect and certain analysis for analog descriptive codes etc., that could be a connection for assails code, assails code for multi-channel to connect the source so you can use that.

(Refer Slide Time: 1:25:29)



For manual test setup, this window tool in that one we use for the automatic test setup action is been that, there is a real time with this use the IDE and the house file form a normal testing. So

that's about test setup, test setup specification how it is prescribed in the automated test setup, and in manual test setup how it is prescribed.
(Refer Slide Time: 1:27:12)



Similarly we can have test software build, tools information and the tools option we must use, we know the test software build is something like a fresher build. It used for test software inside the embedded target system, for this detail called tools version or and any special features in that we sold be taken care it will highlighted during the instructions and some of the tools may require to be qualified it is called tool qualification or tools qualified data all the tools want to be qualified tool qualification in the senses for examples an aerospace project they look for a stringent test process they look for a stringent guideline as per the DS of course defenses standard.

So that only specifies as a tools that has be used for testing or embedded software should have a qualified report that means the tools have to be proof the tools generate a integrate or a defect producing tool or defect producing machine to that is to be refer to the position similarly we have a software develop procedure that resulting the earlier sections downloading instructions procedure along with the any of the operations.

On the test board so this will be part of the first category so this the part of the test cases suppose if we have the test connections but all this have to be part of this producer that means which we are use of anything that are related or relevant to this questions so that is an example a test cases example of a test cases especially the document we been identifying the specifies it applications under table and the tool of any other functions, finding or disposing.

in case on environment automated and how it going to be executed and overall options for testing behavior or any manual testing then only target code etc. then we have producers for the using automated testing nor manual testing in the end we need to have one special importance about the tools that is about test cases we have so that is looking test case design and producers can be specified or maybe we will continue this session at the last we have the test standards or which interfaces in the table

