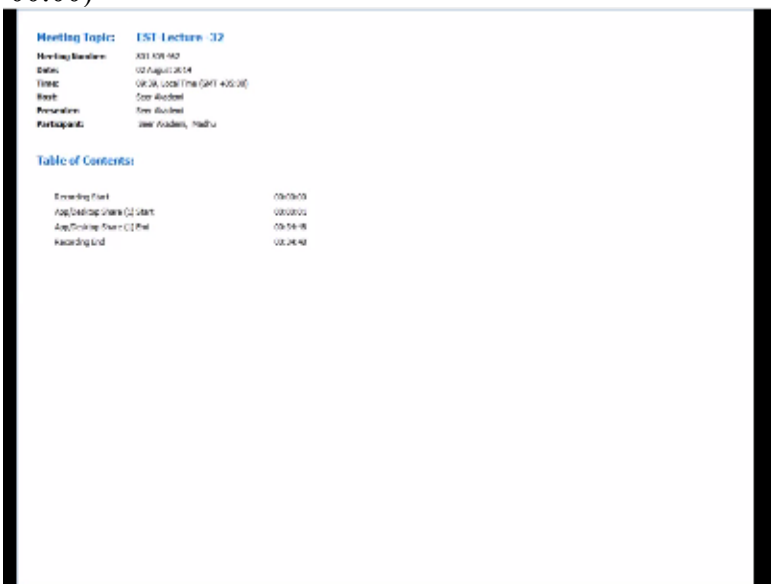
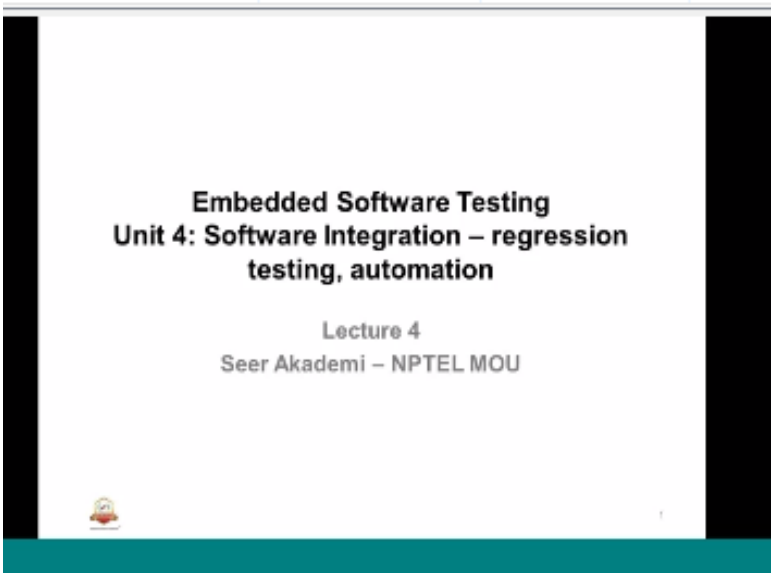
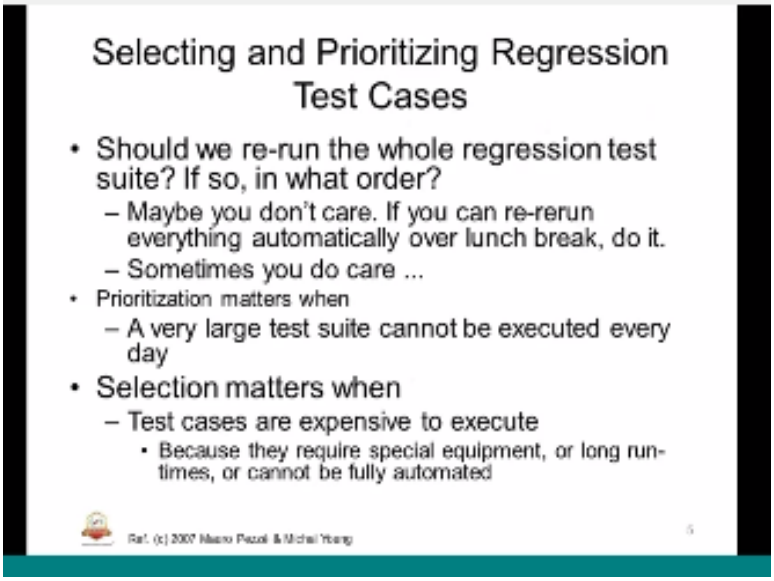
(Refer Slide Time: 00:00)



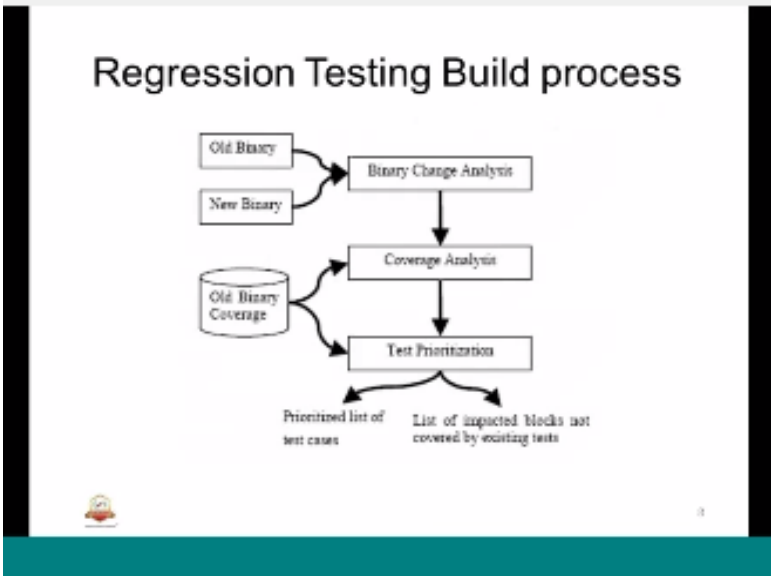
Thank you for joining the next session of embedded software testing.
(Refer Slide Time: 00:05)



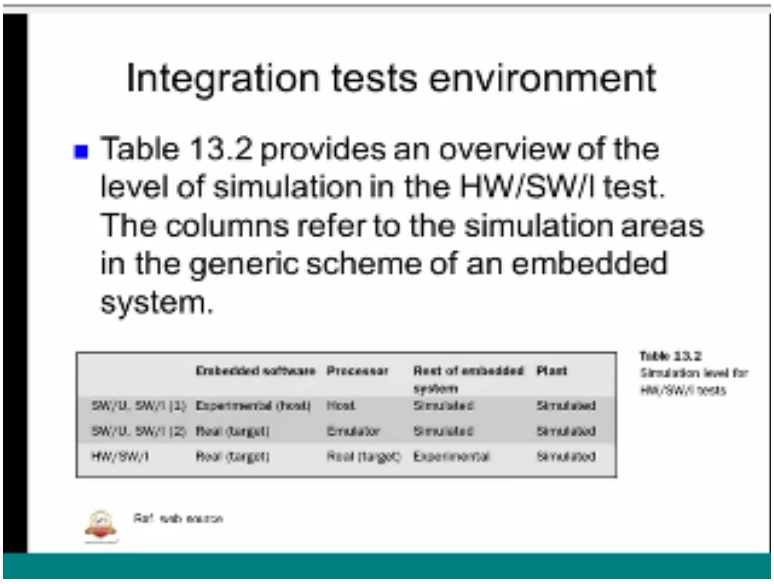
This is the unit 4: concluding session on suffer integration – regression testing, automation and study more about the selecting the regret us test maintaince.
(Refer Slide Time: 00:24)



Regression test maintains regression test build process.
(Refer Slide Time: 00:28)

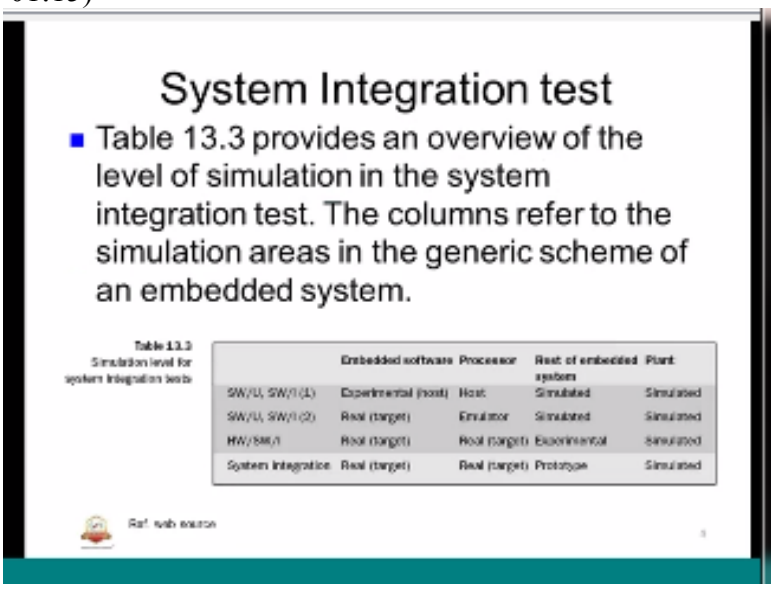


What are the steps that are involved for regression test will build process test strategy and we will try to recap the session unit 4 all the sessions. So before that you just to equip walk through of what we have understood in a higher session.
(Refer Slide Time: 00:46)



We understood about the ingression test environment like it could be hose based processor or emulator or it is an actual target. So we have a different task of introgression that software unit level software ingression level software to hardware level, to hardware level to software level. So likewise we have an ingression test in environment defending.

(Refer Slide Time: 01:15)



And also that is another aspect of integration test, the system integration test which involves system level where we actually we used the real target. As –well - us the customer embedded system suddenly surrounding that also will be a real one I mean in terms of reproduction one. We are use actual the module, are product module of the other embedded systems. Which are requires for example re inputs to regard we going to regard with the actual function generator or

whatever we will not use the tubs are any drivers for driving the values. So that is where system integration will take here.

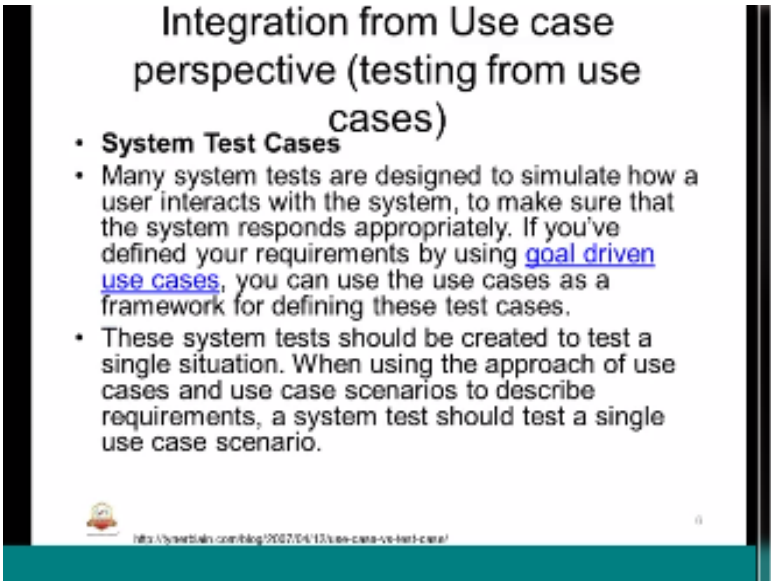
And next to that is an actual system plus which is on the production board and also it is an expectants test. There we delivered the finalized product to the customer with the sample acceptance as per the specification with require.

(Refer Slide Time: 02:20)

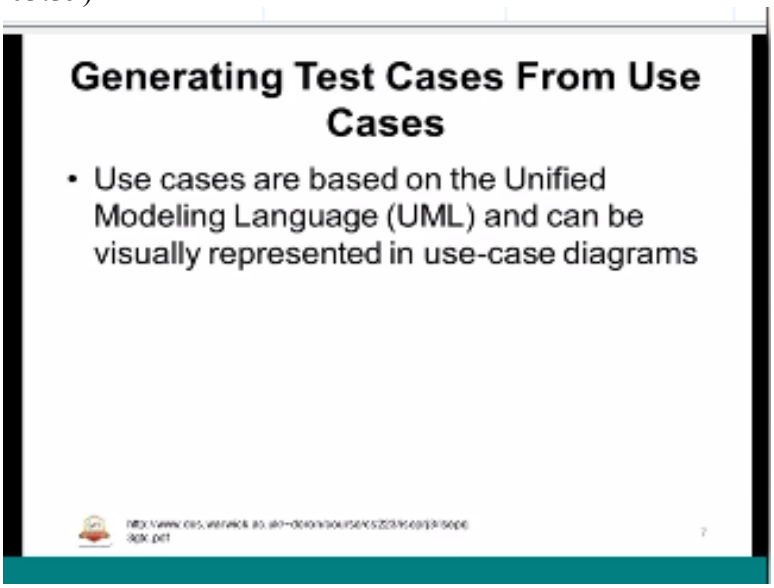
So that typical behind the aspects for the or the industry they follow basically now so studied about integration from use case perspective like we have a user perspective understanding the embedded system and lot of synergy have return these synergy can have multiple or single test cases research cases as usual we will test base procedure scripts and execution fallow of the basically the absolute behavior of the system will be perused from the users perspective. That is where the use cases are already used.

If you have a modules act structure, returns achieve that it can be tested from the use case perspective it is good to have an integration done from the use cases. What we do is we first describe the users are the actors and describe this scenario and for each of this scenario will write the test cases. So this called us gold driven now use cases.

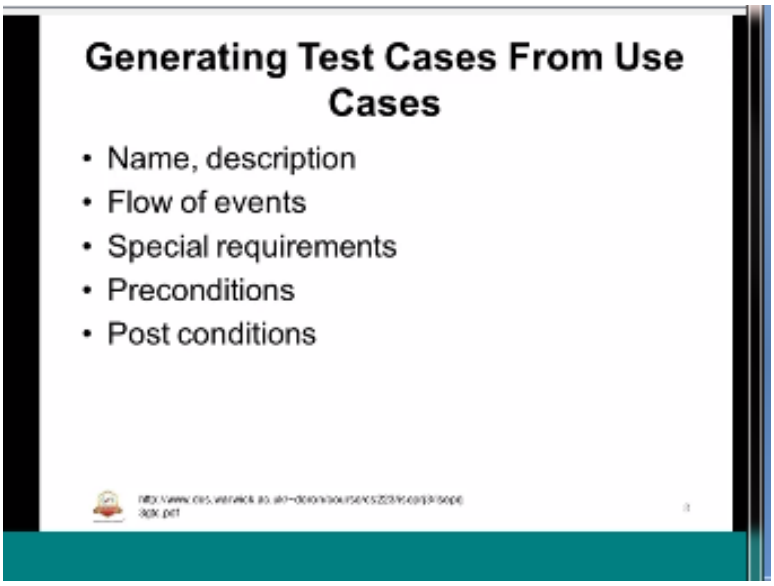
(Refer Slide Time: 03:27)



And generating the test cases basically there are lot of tools are available.
(Refer Slide Time: 03:39)

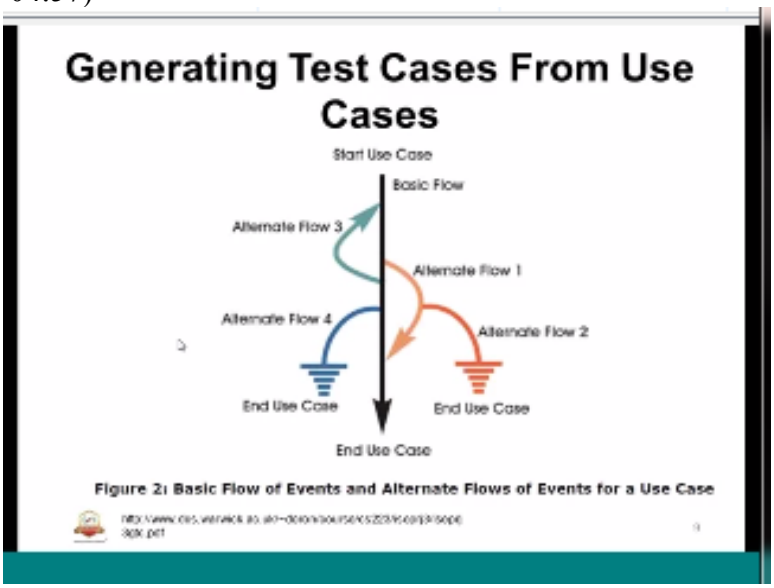


First we develop the use cases with the help of tools such has UML unified modeling language basically it is a modeling language were we product and model the embedded systems in terms of various diagrams but for developing test cases we represent use case diagrams. We will try to understand few examples of use case diagram to this so this not part of the scope but I tester to explain one or two slides.
(Refer Slide Time: 04:12)



And typical they use cases will have name of the user or the tester or the actor it is called has and description of what the use case is going to what the user is going to integrate how is going to integrate. Next one is flow of events, what are the flow events? That user can take then any requirement that it is getting addressed it special requirement that is the addressed and free conditions for doing that flow. Similarly post conditions after that synergy is been completed.
So likewise we are going to have use cases and accordingly we are control generate in the re test cases.

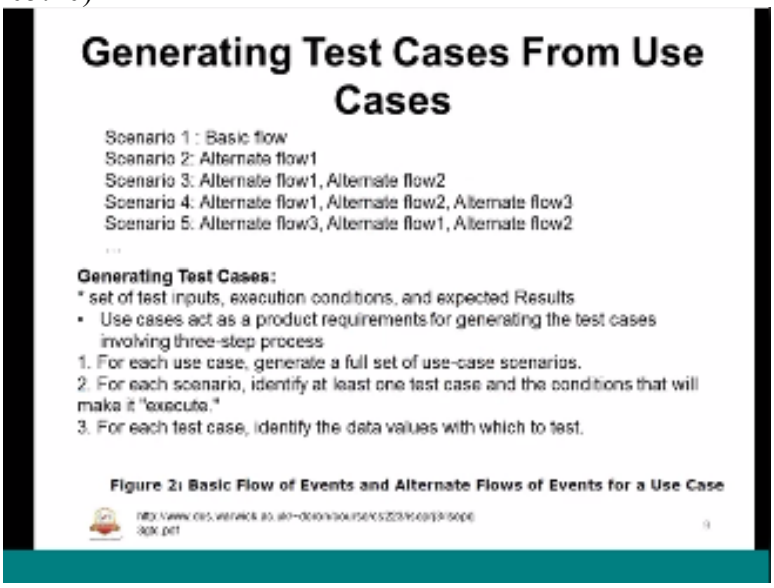
(Refer Slide Time: 04:57)



You can see the example here figure what we are seen it is a bascule of events and alternate flows we can see the red mark the once for a use case. Here the different this is grounded, grounded inherence is going to end there, end of this case they mark this way the typically the

flow basically flow that the alternate flows will be drawn like calculate flows 1,2,3,4, depends of the type of system like to use for each of this flows we are going to how prospectors one of the prospectors.

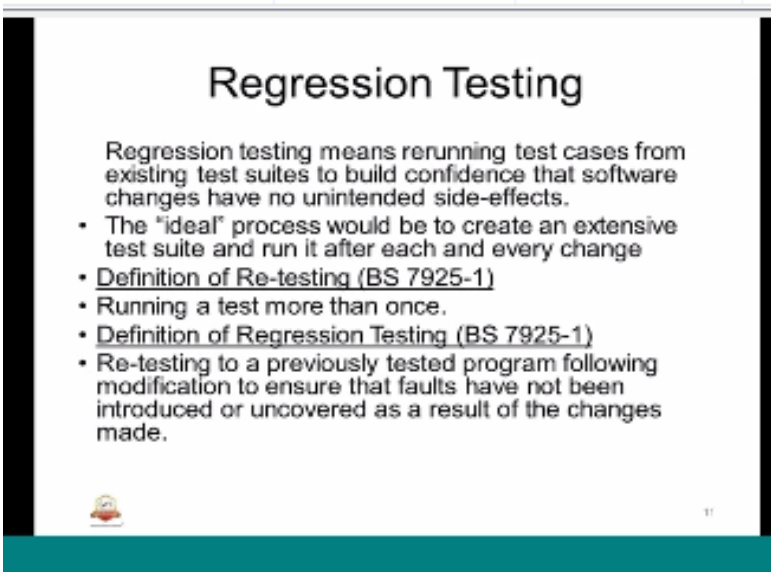
(Refer Slide Time: 05:40)



We can example 1 or 2, 1 to 5 test cases are return the test cases can also have multiple flows involve because we need to achieve to reach the part that is we need to generate. So basic steps for generating the test cases involve for each use case generate the use case of arrays for each in aria identify at least one test case and conditions that will make it execute. And for that is case identify the data values.

That is objects for the inputs that are going to be used. It which we are going to test it are get data values can also here the condition with which we are going to vary and regard, and the data values also will have a expert output us well as the actual or put that is going to be provided when will going to that is going to be output when we executive test case then after use case example.

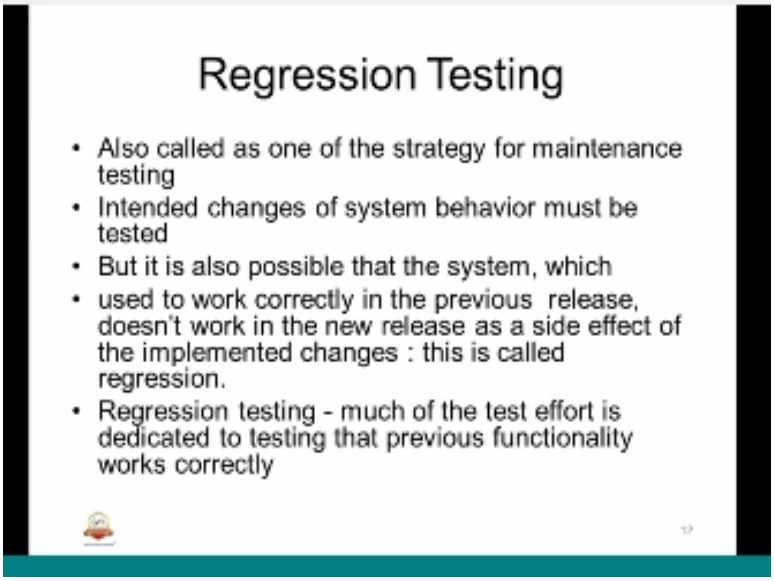
(Refer Slide Time: 07:01)



We had a studied about regression testing, regression testing can be divided has a retesting and regression testing. Retesting is testing again and again, regression testing is retesting on a previously developed programme with the check source to make sure that the false that war the before or clear this time. This is where a regression testing is important specially we uses same test root the requirement will be same.

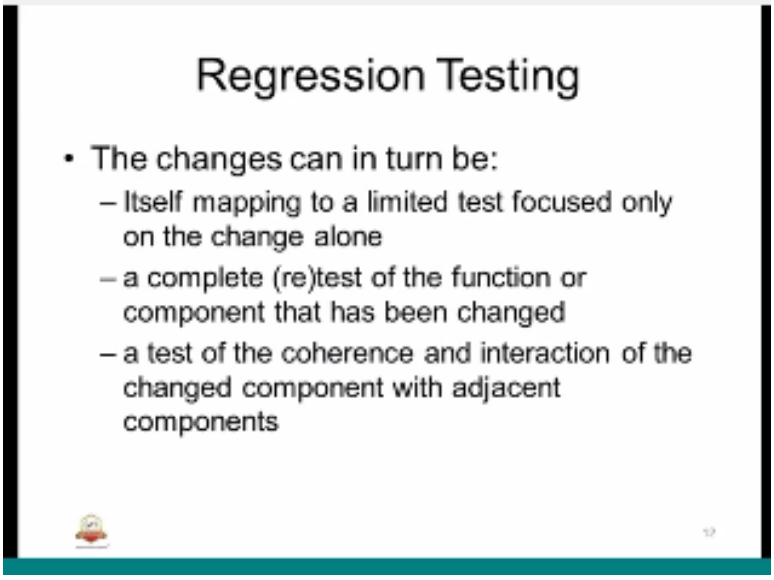
On the thing is the code which add earlier, works have been fixed and we have a new option of the build on which event to re execute that is where regression testing which dedicated.

(Refer Slide Time: 07:51)

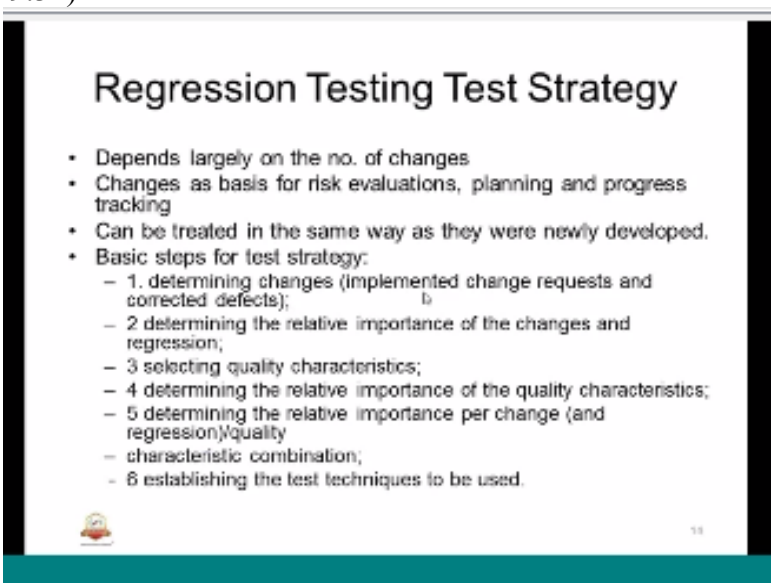


Also is called as one of the strategy for maintenance testing we try to understand what maintain testing inflow testing. So the changes there are there in the systems are to be tested with the help of regression test sort. But the chance for that along with the 6th test would there be an impact on

the other test also which could fearing now, which ever passing earlier. So we need to choose the regression testing synergic search a whether the analysis and impact of the text us up delivered on the existing build is to be redone.
(Refer Slide Time: 08:46)



So the chances can be of free types it is on the retagged components it self's specific changes are specific line or a small systems, small search system excreta. Are complete test retests of the function we will have to do it to measure that regression is taken care. Already third type is coherence or the interaction of the changed component with the adjacent components of the surrounding models was the imp actives there so all these three areas we need to take a care.
(Refer Slide Time: 9:32)

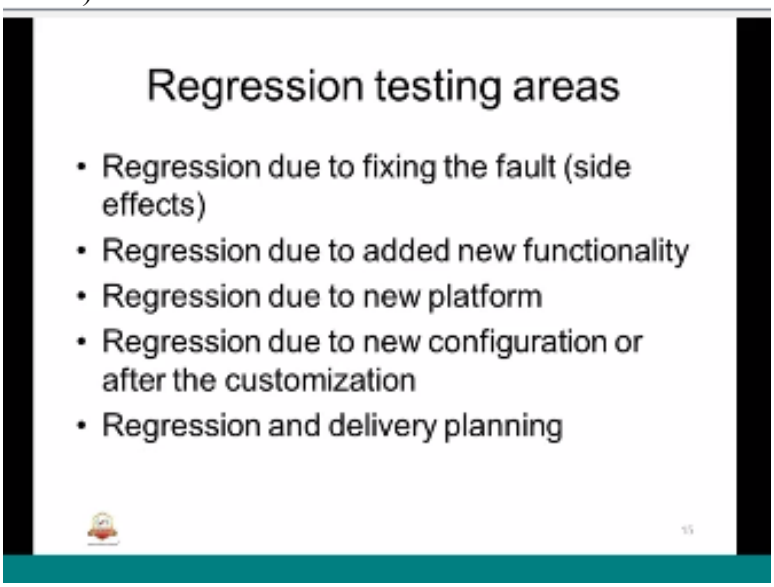


The regression testing test strategy definitely depending on the number of areas and the kind of changes so we need to evaluate what are the risks that are going to the valuated and how we are

going to plan when progress tracking of the embedded system so for regressing testing. And sometimes new wire for danger test as if the code is being the newly developing. So test strategy is to change again implore.

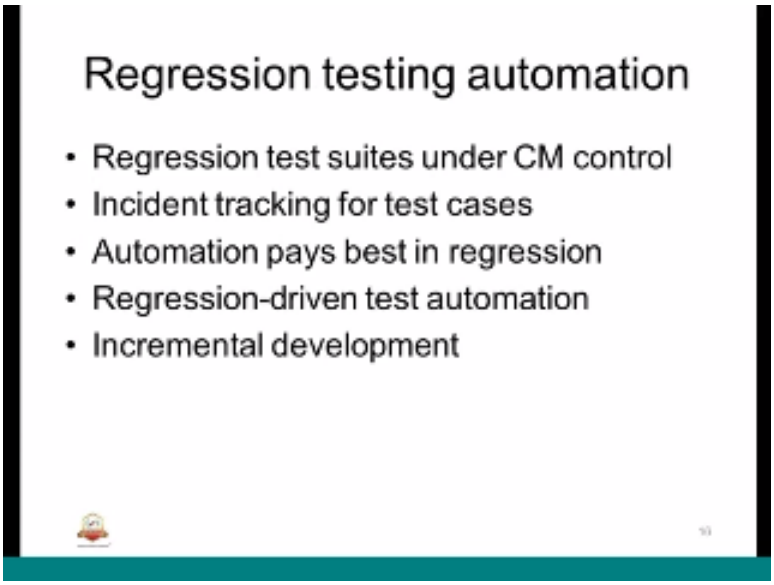
The deterring the changes let determines the import of the changes. And regression selecting politely characteristics deterring the relative importance of the quality characteristics deterring the relative importance for each change of the quality characteristics combination and establishing the test technique which has used for regression testing.

(Refer Slide Time: 10:32)



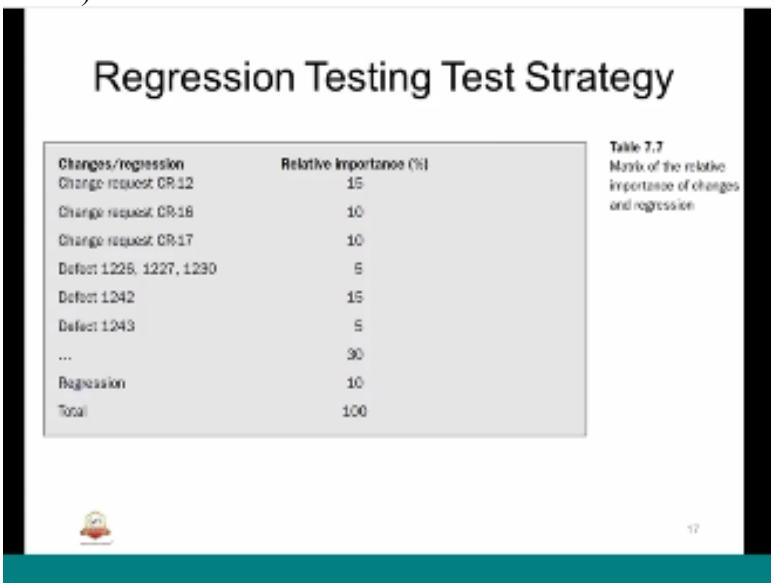
Regression testing areas are regression due to fixing the fault regression can be for added functionality regression can be on the same build same requirement, same specification that fixes on a parroted platform for a new platform. Regression can be due to new configuration or after to the customer machine of the software. Software has not change but it configured for a different values with the different customized situation. Regression and delivery planning that also need to the plan.

(Refer Slide Time: 11:11)



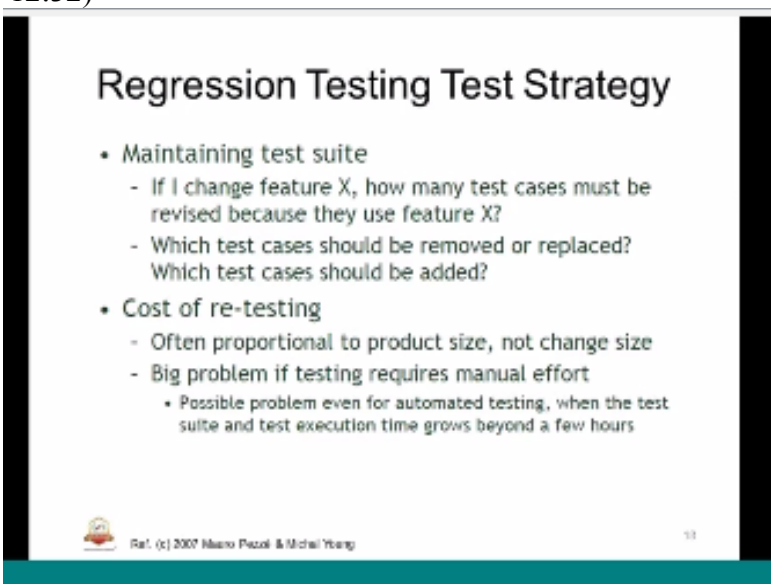
Regression testing automation is also one of the important are where we are going to develop the regression test suites under conflagration control. So automation can be developed or a period so that every time we do not re execute sudden bundle of a test manual day so we automate those batches has the automated script excision so that regression can goes smooth. And it is going to be incremental development.

(Refer Slide Time: 11:52)



And there is regression test matrix the relative important of changes and regressions the changes could be due to change request that are accord during the fixes are re fixes what are base the changes could be due to defects the identified during testing. And the importance could be 30 to 40 %, 10 to 30 % based on the type of war system that we have. So that is over the relative importance of a regression there will be working note,

(Refer Slide Time: 12:32)

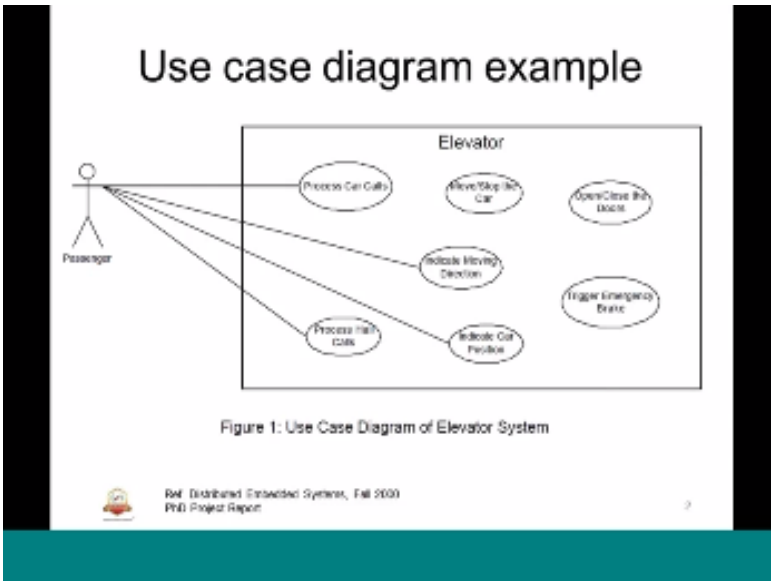


The basic problem of regression test is maintenance test would and cast of re testing if the feature X is change how many cases we have to execute for that feature or impacted the X. Because, the revised X could impact on the other areas as well. And what test cases can be removed or what cases should be added? All this should be taken care. So will may have a test root or the test laugh of the batch files but, ignore batch files all may not be needed because we are wasting unwanted things.

We need to choose so that is the maintains so it is going to cost some a fault and cost of the retesting of an professional to product size, not change size so what are the change we have and what is the total size of the product. So that drives the cost of the retesting. Basically it require some anal effect, for doing the regression at times why because the changes are too much and we cannot offered to other same side of a automation sometimes the changes are very mini well so entire bunch into re execute.

We may have to choose for tailor the excising the automation aspects so that is what we have studied about the regressing testing and the areas in the type of them today session will try to understand the in countries of the previous one.

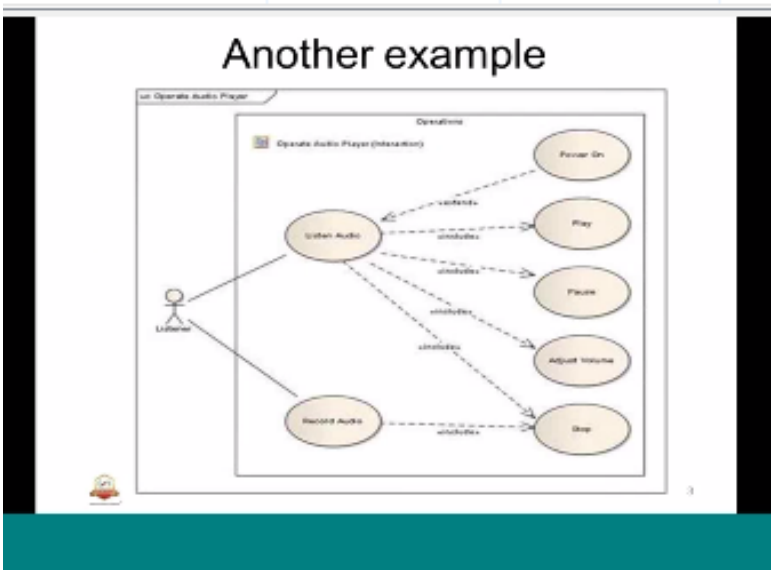
(Refer Slide Time: 14:16)



Use case diagram example just briefs it so this is the typical use case diagram you can see this diagram is drawn for an elevator system. You see use case have a actor and this elevator is a functional block so there are different figure we can see there all it vole process boost of the car open closed the door indicate moving direction process all calls indicate car position regard emergency break.

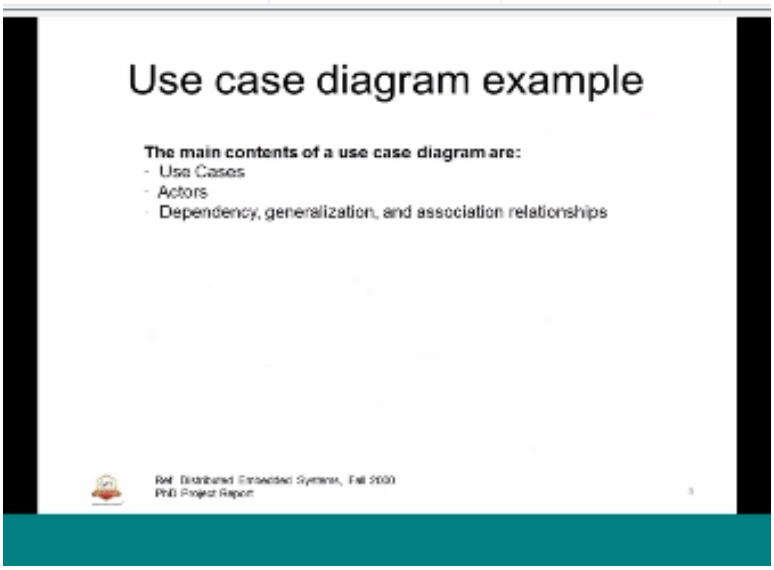
So all this are different in arrays that are used and this can call for different test cases, so that is how we are going to have use case with the scenarios actors and a function blocks. And each of the scenarios need to the explained that operatory. So that teat cases can be drawn easily.

(Refer Slide Time: 15:26)

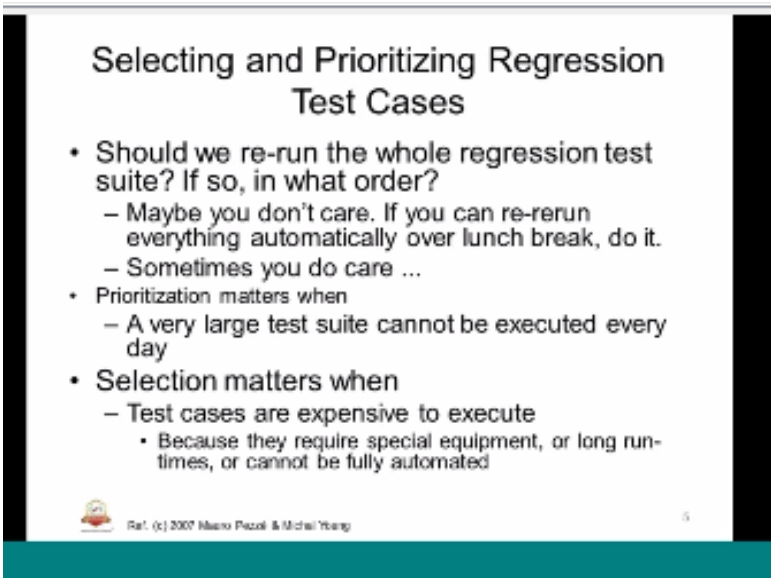


So see another example you not sure if it is clear that is an operator or user, so this basically apparition of audio you can see user can do a designing audio or be recording audio. This basic

functions is going to act and this listing audio can further add the different operations are use cases scenarios something like a porn, play, pause, adjust volume, stop. Similarly the record audio can also have these stop etc.
So this in arrivers for each of this arrives after saying this it draw test cases right so is important to have such models having the use cases drawn so that the entire life structure for example convey tested, which help of test cases. So it is easy to test right so properly try to develop a test I do not know assignment for this particular use case one of the session in the feature lane. So let is how use case diagrams are useful and to be used for integration testing okay.
(Refer Slide Time: 17:01)



So basically the main contents of the use case diagrams are use cases it self's reduce the compliant thing with arrivers then the actors we can see it can multiple actress also because multiple users are user small means does not mean that actual end user it can be any function block which is calling sub function or if there is an input object regarding condition which in call for this flows.
So that is although there is actress or defined. And the third one thing a decency, generalizing, and association session how will go to be associated with the diagram. So these three aspects are the contents will be there in the use case diagram with help of this =test cases will be drawn.
(Refer Slide Time: 17:58)



Okay so coming to the regression testing, regression test cases selecting and prioritizing the regression test cases these also one of the important point in the regression testing. So the question is should we re-run the whole regression test whole in what order. The order is also is very important because we have different test cases flown in different manner so and the use that time referred and conclusion and the problem areas.

All this matter in terms of realizing the test we cannot offered have a high complex testing to start with why because we will spending that in the beginning early too much of time and later stage has is very difficult to pick up the small once. So we need to have a balance of worth the complicity as-well us simple one.

In order to make it in good order of regression test So for this question answer could be something like maybe we do not care if you can raven thing automatically over lunch break to eat that means you have a confident that entire bunch can be raven without any issues, without in stockade worth time something like 2to 3 hours then better to go for it for the entire regression suit for the bundle.

But sometimes it is not true we will take here like we will break up the exercise the test root into such a test root that it is going to map and realized the regression areas were with use earlier and that is regard in the current execution. And accordingly we need to paradise that different types of regression test roots and test fineries so prioritizing matters when a very large suit cannot be executed every day.

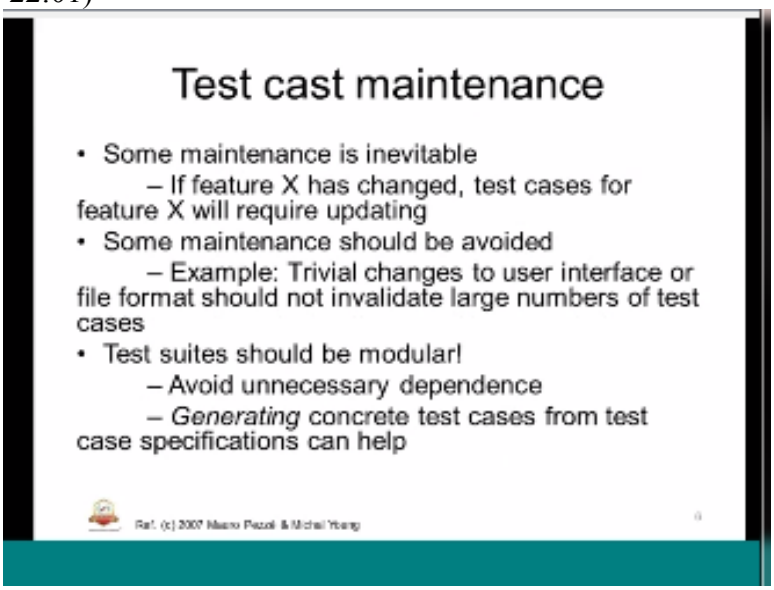
So we cannot have everyday operation in terms of test root execution as a batch integrating testing will to prioritize. What is the prioritize? Functionality a, b, c, d suppose where there and

we have a for each function fourth suites for a, b, c, d then we need to prioritize depending on the type of regression.

That means high complicity and that functionality is very key and important and we need to prioritize that have a higher one. Where other functionality has is like smaller functions or routine functions which present requirement attention can be having the regression priority. Selection maters when test cases are expensive to execute that means if it is going to occupy the special equipment or lie senses or is going to cost more time when the selection is also is very important.

It is not just in half to have to prioritization only prioritization and selection both matters for regression test cases. Because again there is a cost in world affect all this set up a is complicity and dependency all this maters in terms of regression testing execution. The test cases are excessive to execute because there require special equipment or long run times or cannot be fully automated. So this is been referred in the one of the more of send McLane the book so let we as told that selection and prioritizing of regression test cases are important.

(Refer Slide Time: 22:01)



Okay so another aspect of integrating testing or maintenance or regression testing is called test case maintenance. Okay test case maintenance is another aspect which has to be maintained for a periods suppose the product is for two years and 50 % of time is dedicated for testing definitely we need to have a an allocation and dedication of this aspect called test case maintenance were a test cases need to be maintained because this is the chance that this test cases have been revisited upon sub division of this software changes in the requirement or customer confidence or multiple platform excreta.

So all this can be taken here with the help of test case maintenance activity so some maintenance is invertible if feature excess has changed test cases for feature case will prepare objective. So it is not just in half to have regression test analyzed used is also important to have the understanding of what got modified or added for change and accordingly we need to modify are alter the test case. Though need to have a test case started from the scratch because already we gave a pram or the free conditions and all that elements.

We just we are going to maintain the in terms of updating it some maintenance to be avoided example trivial changes to user interface or file format should not invalidate large numbers of document that is the small letter like edges some change error test case maintenance of course it is good change at general I say is the name of the signal is mention wrongly. But, test cases were so there is point in doing such maintenance.

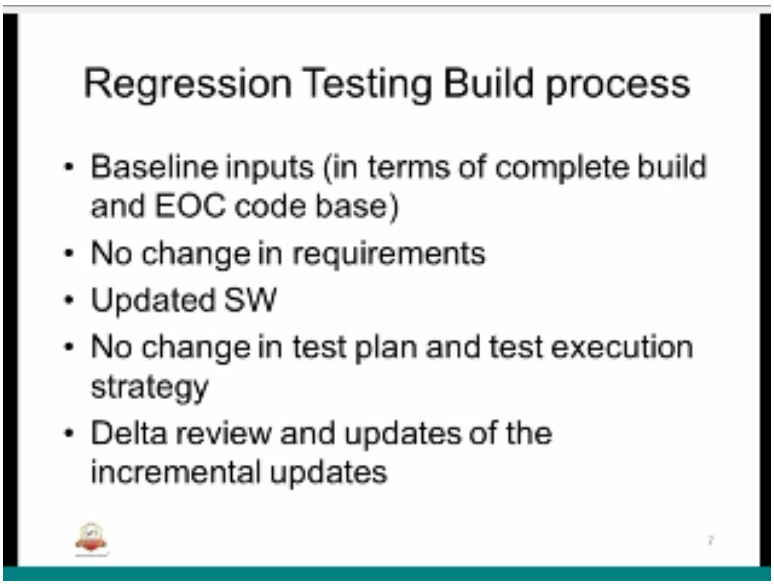
So you just document and recorded somewhere or make not and we actually look into that any other impacted changes you can take up this as well.

So that this can be prioritized and updated when we are going to do it so some maintaince we can avoid for the help of doing the maintenance with doing it. That is the meaning of it, that suit should be modeler that means any dependency we should try to avoid as much so that, that can be independently executed.

Generating the Concrete test cases from test case specifications can help means for the concrete test cases we need to have called has a golden test case. So important the word the using embedded software testing If you have these the minimum test case which is used has a golden test so that this work always.

So if here something stack, Something got issues and we do not know where to go to so what we do is will pick up this golden test rule for test case will try to pause it on and execute so that just is find there is some other issue specific to whatever we are trying to attach in the other test case. So this will high fill that the problems that we have so that concrete test case are very important in maintaining the testy root or test cases. That is where test case maintaince in product is.

(Refer Slide Time: 26:13)



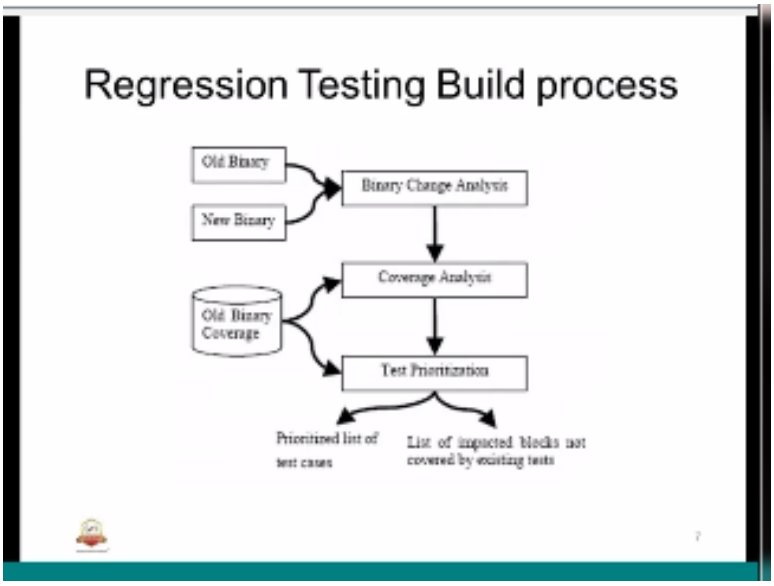
The next one is regression testing build process how are going to have the build processor done for the regression testing. You know what is build? Build is basically complexion linking and development of the software. Similarly we have the build process for the text equipment also are the test scenario or test script also. Again there going to the complication and logically grouping the keeping in to a proper players

And making sure that license and LR all this are part of the build process and to bolts, bolts points for the same is arsenal inputs in terms of complete build and execute of launch code USC that is called us the base available. Because which you cannot do a reference test right. So base line inputs is one of the important inputs no changing in requirements update software that was builder changed.

In test plan and test execution strategy it still same but we are going to do a regression on the updated inputs such has EOC executable lines of code. The build then the last part will be delta review and updates of the instrumental updates so we need to review what code changed what is updated and we are going to regression test accordingly so these are some of the build process they going to have for the review centre .

So we start with the data input and we going to analysis the changes and we are going to execute the software. With this baser input so base line inputs will be process through confiscations input basically. Confiscation control will do through various tools such has previous demesnes lots of tools there. We will study that in the next test management of competition testing.

(Refer Slide Time: 28:24)

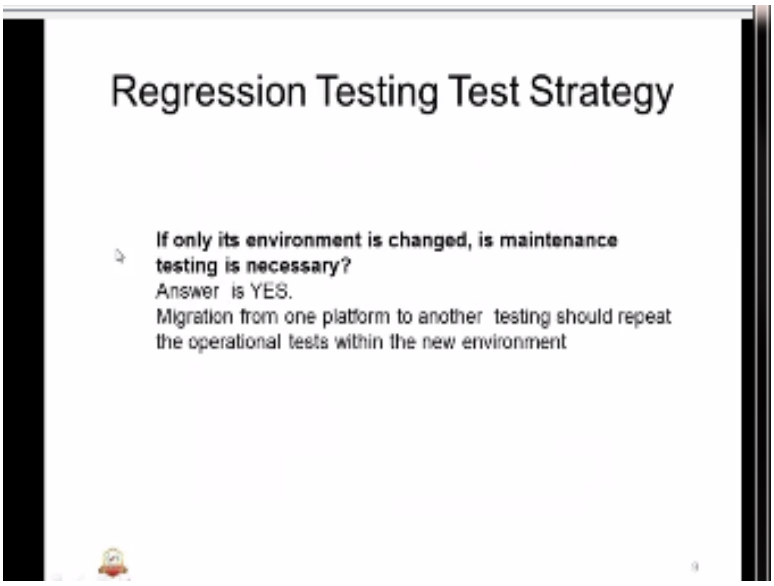


One example you can see diagram here that is in put for regression testing build process you can see a regression of old binary and two binary. Will be added to the analysis binary changes has is this basically a process block were basically analysis of what was there before old binary and what is the change that is done in the new binary. Binary here means the executable code is the object that is going to be tested against. And we are going to do a coverage analysis of the changes that are there.

So that what are that interacted areas that is being covered and how are going to be test it. And for the coverage analysis they could be used in old coverage, old binary coverage or the old whatever we have executed earlier with our suit that will be important. Then we are going to prioritized they suites are the regression test which we are going to start and prioritized piece of test cases were going to come of the test of a impacted blocks not covered by regression test that means what are the impacted block which are not covered with the regression test there also will comm. Has an output of the test priority activity.

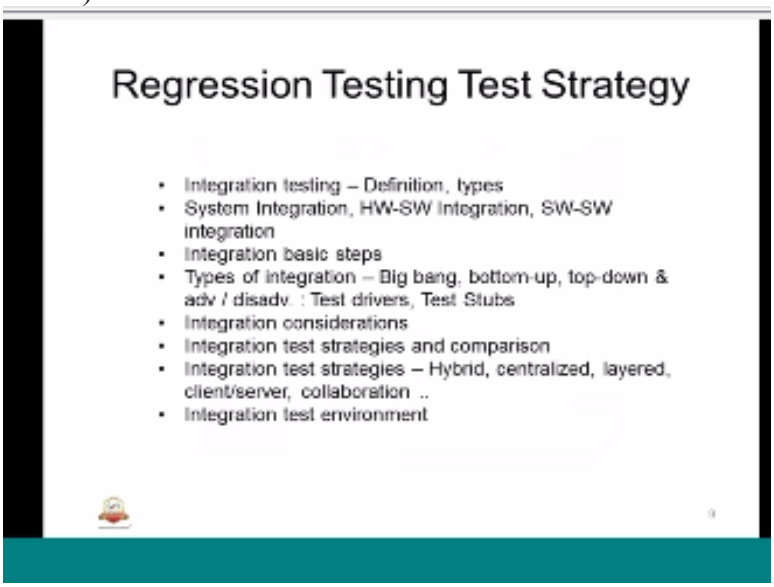
So this will be a typical test building the process of the reversing testing okay having understood all these test case maintaince selection and paradise in other one question is that if only environment is changed we know that the environment could have in change for the new thing he is maintenance testing that all.

(Refer Slide Time: 30:31)



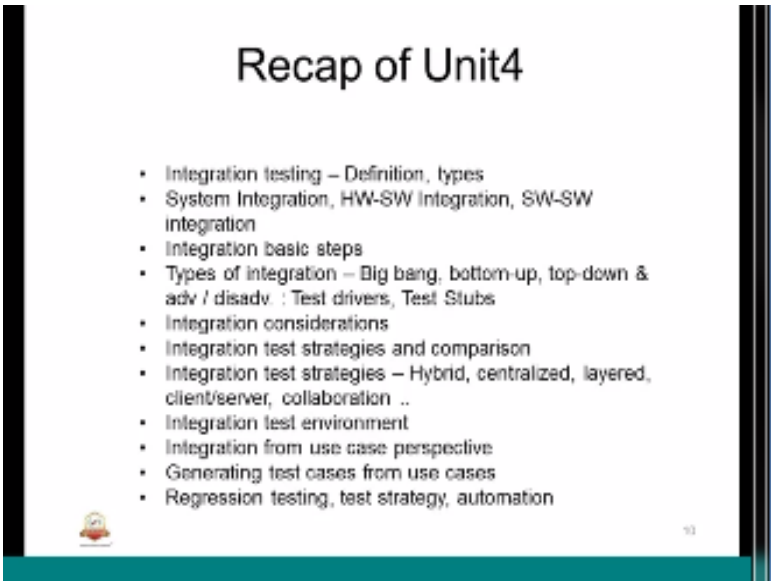
The answer is yes why because regression for 1+1 to another question another testing should repeat the operation test with the month. These execute everything are maintain the testing are regression testing only thing that we need to take carious the operation test which are very much coupled with the target platform are the platform change has with the new environment at to the considered to the taken here so that is where the regression testing institute adopted.

(Refer Slide Time: 31:09)



Okay so we that we session of these and we try to recap this unit of regression and intercede testing so what are the topics we studied about regression testing definition.

(Refer Slide Time: 31:21)



And types of integration we understood we also went through system integration hardware basic. What are the integration steps that are want types of big bank advantages or disadvantages and we also knew about were are going the test drivers were going to use test steps ,We can complain both of them there is one the complexity and the test strategy that they going to have ,and also integrations considerations.

We studied what are the steps that we are going to consider in terms of hardware, software environment emulators etc. Integration test strategy comparison we did with the table kike in terms reusability and availability etc integration test strategy how all going to have a integration test strategy drive it could be bottom of top down mix in terms of hybrid test strategy in terms of OSS operating system are skilled centric we can use a centralized approach also we can protocol are network sort of a testing it help of a layered a integration testing approach we can have a database, application.

Embedded application testing can be done with the help of clients server integration test approach you can have collaboration mechanism layered different sub-system are going to be collaborative while doing the integration testing, then also we understood about integration test a normal were all the environment were all the environment in terms of the emulsion familiar from actual target.

Or a hash based testing are define and used and also we understood about integration use case perspective ,use case diagram example we are going to went through and also we understood how test gets can be drawn from the use case ,and use case scenarios and also we understood about generating test cases.

From use cases the various steps that are involved for developing the test cases or different use cases and the scenarios ,and in the end we are studied about regression testing ,test strategy and importance of automation and last one being the.
Maintains test maintains and regression built cases all this we studied in the unit 4 okay in the next session we will start the unit 5 which is nothing but the test management and defect management so that will be the last unit of the embedded software testing after that we going to have practical sessions and so on questions and Q and A sessions for the embedded software testing with that I conclude this unit 4.