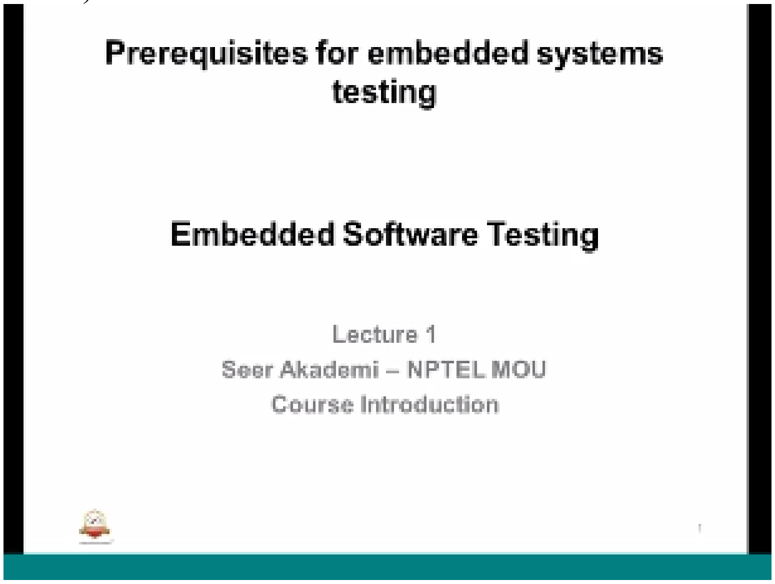
Embedded Software Testing
Lecture 1
Seer Akademi – NPTEL MOU
Course Introduction

Welcome you go to embedded software testing so last time we had a theme some of the embedded software testing.

(Refer Slide Time: 00:13)



Prerequisites for embedded systems testing

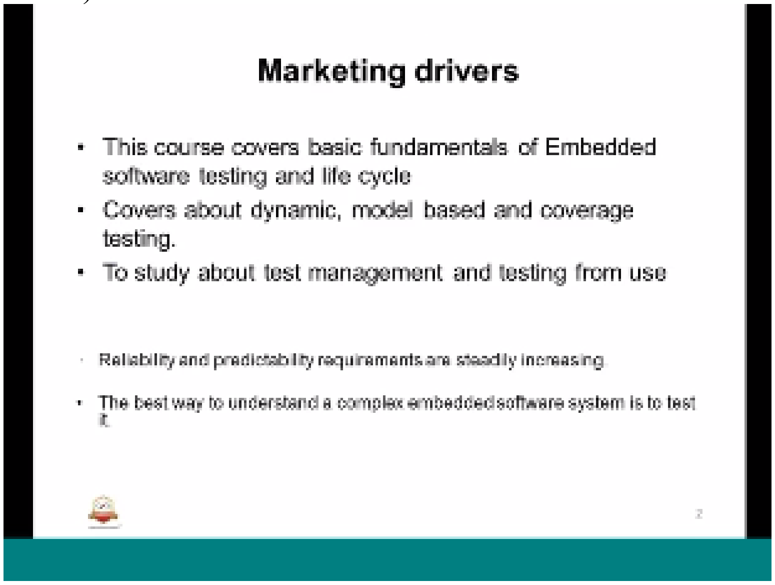
Embedded Software Testing

Lecture 1
Seer Akademi – NPTEL MOU
Course Introduction

1

We had prerequisites for embedded system testing then the marketing drivers' course objectives.

(Refer Slide Time: 00:22)



Marketing drivers

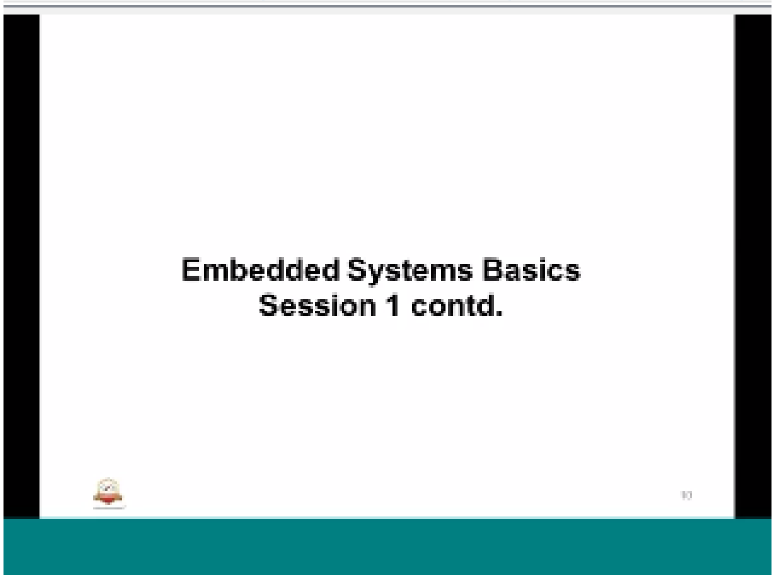
- This course covers basic fundamentals of Embedded software testing and life cycle
- Covers about dynamic, model based and coverage testing.
- To study about test management and testing from use

- Reliability and predictability requirements are steadily increasing
- The best way to understand a complex embedded software system is to test it.

2

And whatever learning come and we had a directly equation about whatever the equation we going to add there are replanted of equations. The software embedded testing.

(Refer Slide Time: 00:50)

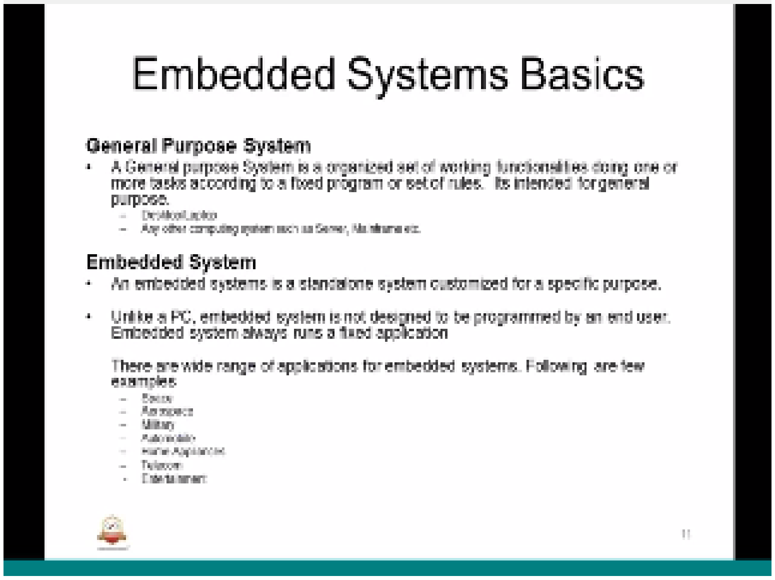


Embedded Systems Basics
Session 1 contd.

10

Okay today it is session I am going to touch upon an embedded systems basics and the session testing what we have discuss last time.

(Refer Slide Time: 01:06)



Embedded Systems Basics

General Purpose System

- A General purpose System is a organized set of working functionalities doing one or more tasks according to a fixed program or set of rules. Its intended for general purpose.
  - Desktop/Laptop
  - Any other computing system such as Server, Mainframe etc.

Embedded System

- An embedded systems is a standalone system customized for a specific purpose.
- Unlike a PC, embedded system is not designed to be programmed by an end user. Embedded system always runs a fixed application

  There are wide range of applications for embedded systems. Following are few examples
  - Space
  - Aerospace
  - Military
  - Automobile
  - Home Appliances
  - Telecom
  - Entertainment

11

Okay so that is embedded system there are two types of system one is general purpose system, another embedded system. So how do embedded system, so general purpose system is organist system is working punctuality more task for a fixed programme or embedded. We will introduce the general purpose like a general computing the calculation for any of the basic $ and excreta. For example we used discord, lab top, for our general purpose for activities or any other computing system it has maintain the computer embedded. These are large computer systems these all accounting general computer system and general purpose system. So now coming to embedded system, so the embedded system is a standard system customizes for the custom system.

So it is internet for doing certain functionality online so unlike general desktop or laptop system. This is mending of the wing sudden of the target gives the system or computer or anything other than the general purpose. That is embedded system I am like to feel the embedded system are not design the programme the user.

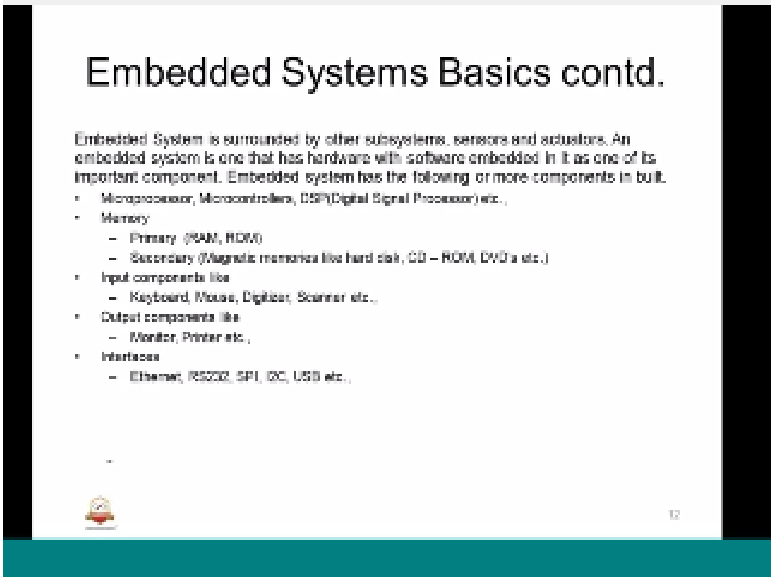
There are PC the personal computer will have a programming capability you can preside user for embedded system touch scale. For the embedded system is always runs functions fixed set of futures with comment of application for embedded system basics. So that is way it is an application for example.

We take a there are range of application for embedded system. The embedded system arrays the military of depends automobile like if you system control unit with fixed now various pass of the car. Then we home control has the TV data excreta. We have a telecom the PDF mobile, tablet and we have an entrainment system that set the set box. And these are some of the range of embedded system purpose. One of the other ways these will have the embedded system refining inside. There is intention to specify

For example here a space embedded will have AVM system mend for control or part is for display for break control or touch of display or like a determined excreta. So we have automobile embedded system or break control or fixed camera of front camera visual system are reverse their monitor systems use of some these example of the guessing residing it could be embedded systems of this demands.

For basically this 6, 7 higher demands discovered to out if the possible embedded system. Which are going to be use with the various commands?

(Refer Slide Time: 05:36)



Embedded Systems Basics contd.

Embedded System is surrounded by other subsystems, sensors and actuators. An embedded system is one that has hardware with software embedded in it as one of its important component. Embedded system has the following or more components in built.

- Microprocessor, Microcontrollers, DSP(Digital Signal Processor)etc.,
- Memory
  - Primary (RAM, ROM)
  - Secondary (Magnetic memories like hard disk, CD – ROM, DVD's etc.)
- Input components like
  - Keyboard, Mouse, Digitizer, Scanner etc.,
- Output components like
  - Monitor, Printer etc.,
- Interfaces
  - Ethernet, RS232, SPI, I2C, USB etc.,

12

Embedded system will have multiple sub system some, something like an error space of activated for in inter causes that are uses. So it can be group of embedded system called embedded system is one of the system of hardware with the software embedded it one of his important component.

So here embedded system means it has all elements of a piece of the box having the special amount of a public application for need so private application to them if make for hardware suppose the motor all should be run nothing but the motor hardware. And the hardware substitutly we need a piece of software that is called embedded software for all this complier in the embedded system and it can have a multiple embedded system. Okay so what is the hard level component basically embedded system.

I usually embedded system will have a basic micro controller array signal purpose, so we will see in the next different what are the embedded system is and RCM, DVDs these are the specific purpose parse. This parses are very much hash of the embedded system array we have a micro desire during all the logic calculation the attitude.

Generally the micro controllers are using the embedded system which is entire for a introducing various devices doing the parse thing excreta. Accordingly we have a DSB which are mending for embedded system highving more put of the calculations are arithmetic or image purpose. This level of embedded system is developed using DSB.

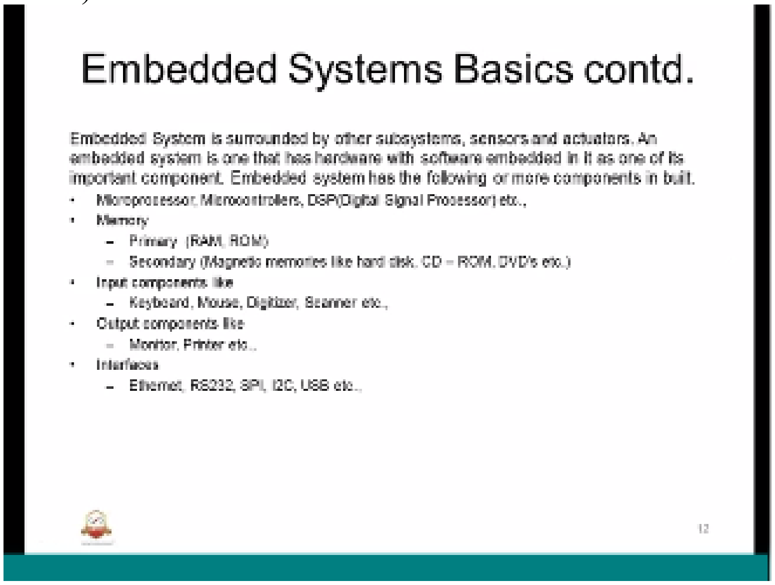
And after the approach the part of we have a memory which is important the curser will interact with the memory could set the data or a programme are instructions memory. So memory order are two types in a random express in RAM and ROM we will not that much about RAM and ROM so it is a will be discuss in the specific of embedded system. So we have a memory the additional memory is hard disks, DVD, telegrams are use this excreta. Those are all part of the memory then we have the inputs array embedded system could be we need camera input RAM and ROM user cleaning some inputs.

It is can be unlock or discussing. So what are the unlocked and what are the dystrophies probably will based we need to the specify end of the requirement used has the embedded systems will have to produce outputs. The action of what is the input to the invariable placement. The outputs could be monitor have been are driving the motor are displaying the some value excreta. This various part of the component of the embedded system

For doing all this we need intersperses, intersperses which will interact to the embedded system. As well of the fictional valued are fictional valued could be other devoice are the user or any other process time. There are various intersperses let us use for example internet for network related embedded system by using RS232 interprets essentially.

The computers having to log in online specify serial traditional interfile when you have a then usually these are interprets requirements. The ended remind the light the curse parses whenever the input intertype and output intertype the intertype devices.

(Refer Slide Time: 11:33)



Embedded Systems Basics contd.

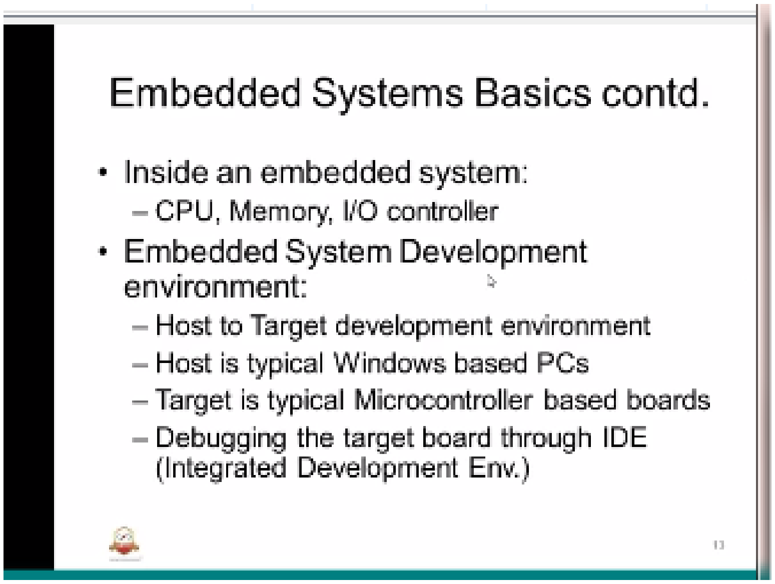
Embedded System is surrounded by other subsystems, sensors and actuators. An embedded system is one that has hardware with software embedded in it as one of its important component. Embedded system has the following or more components in built.

- Microprocessor, Microcontrollers, DSP(Digital Signal Processor) etc.,
- Memory
  - Primary (RAM, ROM)
  - Secondary (Magnetic memories like hard disk, CD – ROM, DVD's etc.)
- Input components like
  - Keyboard, Mouse, Digitizer, Scanner etc.,
- Output components like
  - Monitor, Printer etc.,
- Interfaces
  - Ethernet, RS232, SPI, I2C, USB etc.,

12

The continuation of that we have an embedded system development environment.

(Refer Slide Time: 11:41)



Embedded Systems Basics contd.

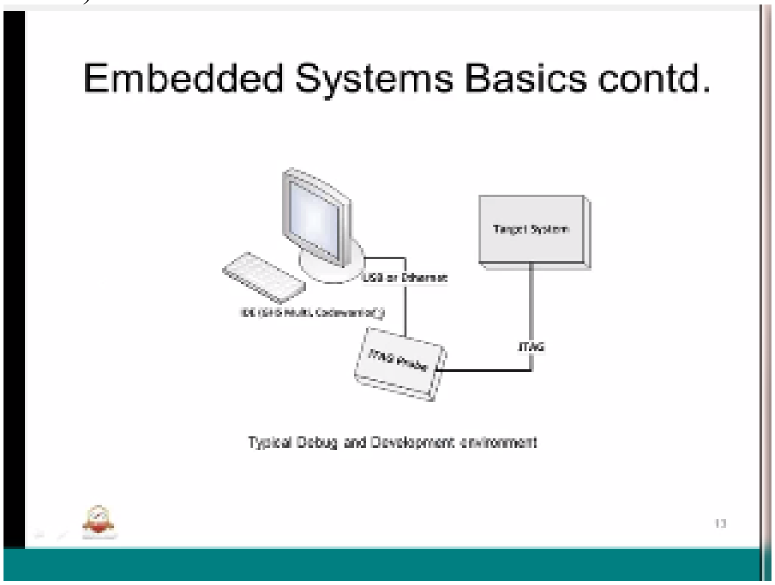
- Inside an embedded system:
  - CPU, Memory, I/O controller
- Embedded System Development environment:
  - Host to Target development environment
  - Host is typical Windows based PCs
  - Target is typical Microcontroller based boards
  - Debugging the target board through IDE (Integrated Development Env.)

13

How it going to be developed host to target development environment can be used, the host is typical window based boards have been connection in the target the embedded system. The target will be a typical microcontroller based boards have debugging the microcontroller could be in of the prosperous are IDE. Environment once you have the set up available we use the debugging that means the there is the tool is to be used it is called the internet development environment.

To the proved we can a used the software called the embedded C or embedded the programme to download to the target board. And they could specific the debugged is has a development level we used the IDE environment. So these are some of the embedded development environment.

(Refer Slide Time: 12:46)



Embedded Systems Basics contd.

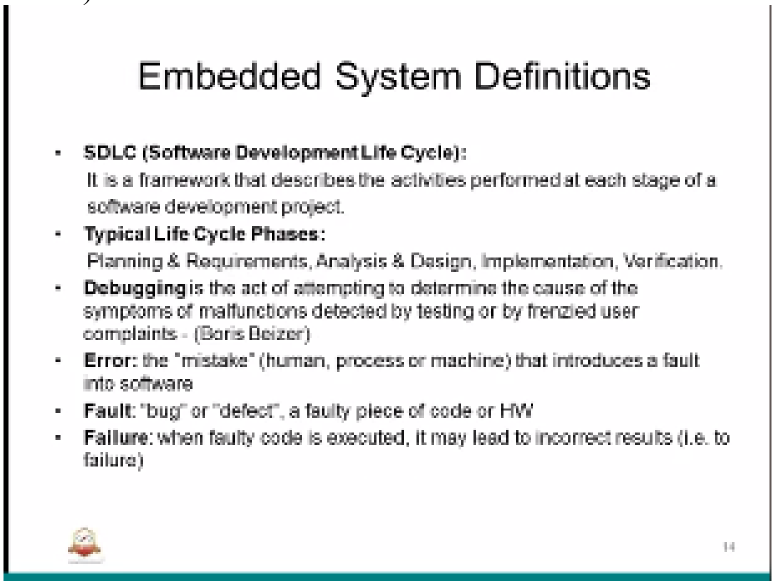
Typical Debug and Development environment

13

So this is the typical embedded system are under of the environment the fixed the lab top or desk top is the pentacle embedded system. There is the intermediate crook box are which is used for interacting with the target system that is called usually edit use card data card video system. Data is most popular one it will interface.

The data system and provided to the home when you get the values provide to the Mr. Andrew and you can task the programme all that. So fixed box you can see a will power that with the target, this is the typical embedded system the other point is normal. Okay we will see some of the basic definitions.

(Refer Slide Time: 13:53)



Embedded System Definitions

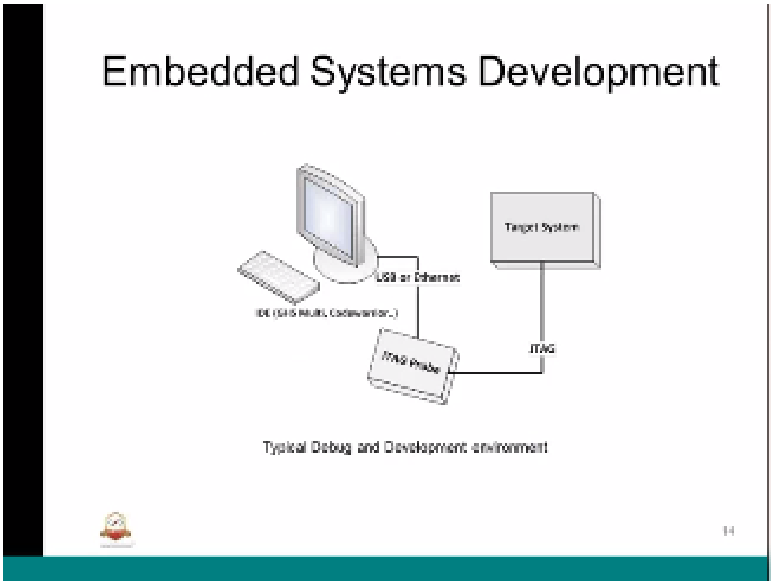
- **SDLC (Software Development Life Cycle):**
  It is a framework that describes the activities performed at each stage of a software development project.
- **Typical Life Cycle Phases:**
  Planning & Requirements, Analysis & Design, Implementation, Verification.
- **Debugging** is the act of attempting to determine the cause of the symptoms of malfunctions detected by testing or by frenzied user complaints - (Boris Beizer)
- **Error:** the "mistake" (human, process or machine) that introduces a fault into software
- **Fault:** "bug" or "defect", a faulty piece of code or HW
- **Failure:** when faulty code is executed, it may lead to incorrect results (i.e. to failure)

14

For using the embedded system the element of definitions software development life cycle to be the main and each under system are hard developed. The life cycle will have different places the requirement the planning parser then you have a design we have implement is that is calling then we have the verification of debugging verifying the embedded system. These are some of the have been software development statement.

And debugging the one is become act of attempting to determine the cause of the symptoms of malfunction detected by testing or by frenzied user complaints. The user definition are body user the entered the statement we will decide the debugging the target board.

(Refer Slide Time: 14:58)



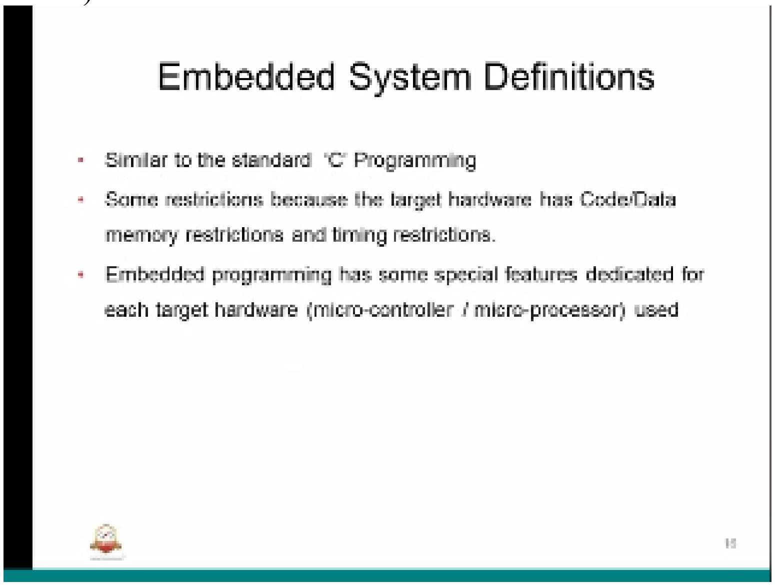
Embedded Systems Development

Typical Debug and Development environment

Are should be what are the programming the use for any mistake whatever the changes there beaver those are the information. These are the development definitions so these three of the testing developing that there are the mistake take or will programming from the develop team they may introduce paten for they are using at the error. And then we are fault bug or defect which could be in the HW software that part of the embedded system.

The defect or system discount then we has a failure when there are fault code is executed. It may incorrect results failure means after we that why the embedded system into the fields it may malfunction are malfunction are executed are that is cannot be except the requirement other these are the result in a failure. These are some of the embedded system hard data base of there. Now coming to programming what are told in the earlier we need the development environment.

(Refer Slide Time: 16:25)

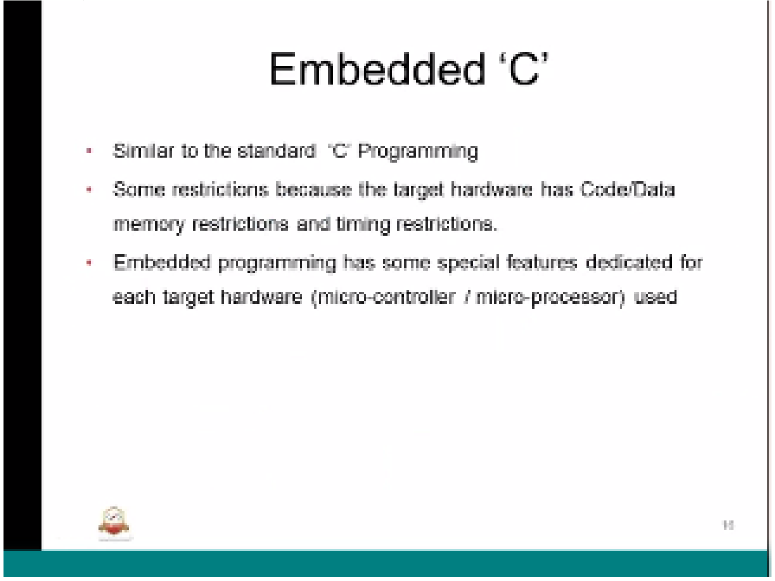


Embedded System Definitions

- Similar to the standard 'C' Programming
- Some restrictions because the target hardware has Code/Data memory restrictions and timing restrictions.
- Embedded programming has some special features dedicated for each target hardware (micro-controller / micro-processor) used

We need to programme to the target system using the development embedded there C or embedded C++ or excreta so embedded C is typical language.

(Refer Slide Time: 16:42)



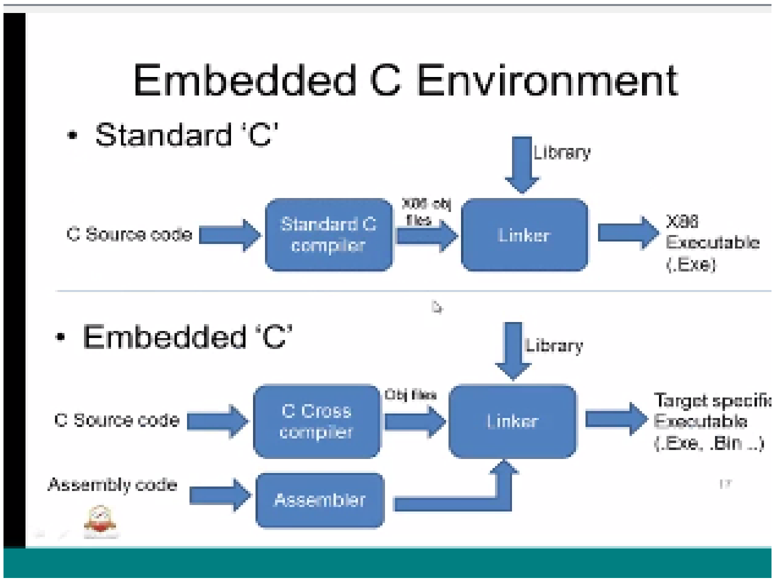
Embedded 'C'

- Similar to the standard 'C' Programming
- Some restrictions because the target hardware has Code/Data memory restrictions and timing restrictions.
- Embedded programming has some special features dedicated for each target hardware (micro-controller / micro-processor) used

They used it is a very similar to standard C like a programming is a possible that is the way it could be used and there is some situation because of the hardware has code data memory restriction and timing restriction. Embedded c the programming some special features and target system will defined the memory or defined timing requirement. So according to that this programming will do and in addition we have embedded programming the special features dedicated for target hardware.

Target hardware micro controllers are control the loud speaker like saying a memory related operation there are special embedded system. So these all part of the feature that is used under embedded system

(Refer Slide Time: 17:52)



Embedded C Environment

- Standard 'C'

C Source code → Standard C compiler → X86 obj files → Linker → X86 Executable (.Exe)

Library

- Embedded 'C'

C Source code → C Cross compiler → Obj files → Linker → Target specifi Executable (.Exe, .Bin ..)

Assembly code → Assembler → Linker

Library

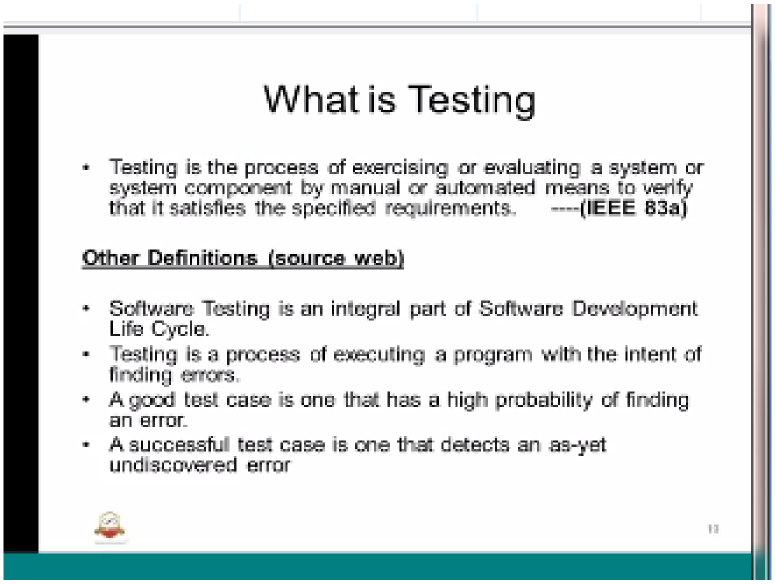
Defeated here, embedded C environment there are two portion which are the first one is standard C for the use it in a general computer like a PC the below in the embedded C use the target basics the development target. You can see the different system of C code complied the PC, Pc nothing but excited the personal computer. If you generate the object file and the object file is maintained using and the linked output will be exsicutable.

That could be exsiccated and the part time bases this is the standard C and we have embedded environment were c sources code is complied. Cross complied what I mean the cross complied means the cross complier will have compiling intercede specific the target board. The target board is having the micro controller not the X 86 it to be any of the motor law or we sale CI based specify that there are cross compiler.

For example Daily complier standard the objects and complier and it get memory it should be bound or it will programme into the target board. Also we have along with the embedded C there are few programming areas something like a book code which are usually developed in assembly language. So this is the assembly along with the comparison which will generate the assembly output along with the library.

And it could be the executable output this will be target board, target board reduces for requirement or field typical embedded environment. Anything on the specific under the basics as we progress and the embedded system Okay in the last session we have discussed about embedded software testing what is specify the processer the reality by automatic are you have to verify that this satisfied the requirement.

(Refer Slide Time: 21:06)



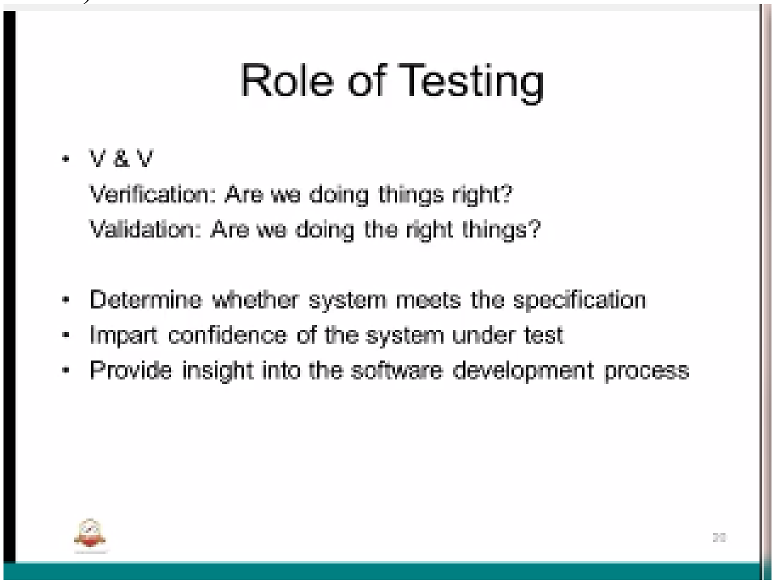
What is Testing

- Testing is the process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies the specified requirements. ----**(IEEE 83a)**

<u>**Other Definitions (source web)**</u>

- Software Testing is an integral part of Software Development Life Cycle.
- Testing is a process of executing a program with the intent of finding errors.
- A good test case is one that has a high probability of finding an error.
- A successful test case is one that detects an as-yet undiscovered error

The role of testing will be verification and validation.
(Refer Slide Time: 21:14)

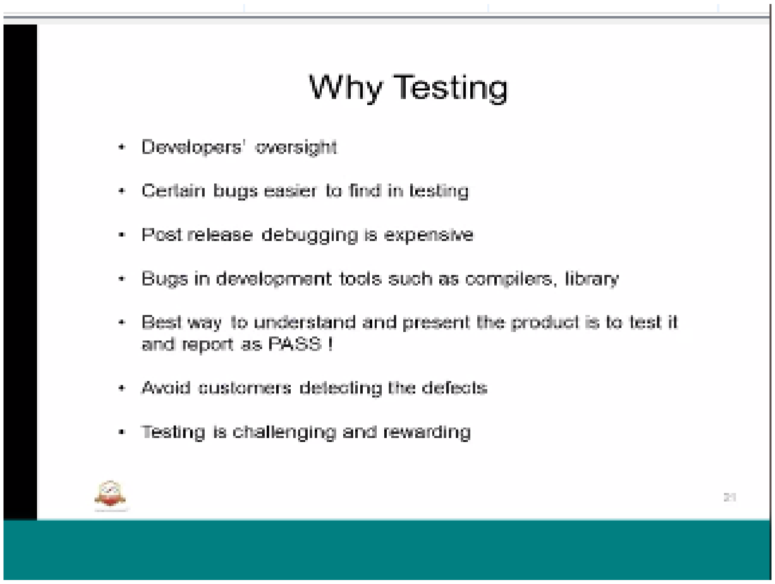


Role of Testing

- V & V
  Verification: Are we doing things right?
  Validation: Are we doing the right things?

- Determine whether system meets the specification
- Impart confidence of the system under test
- Provide insight into the software development process

Determine other requirement impart complies of the provide insert into the hospital process.
(Refer Slide Time: 21:28)

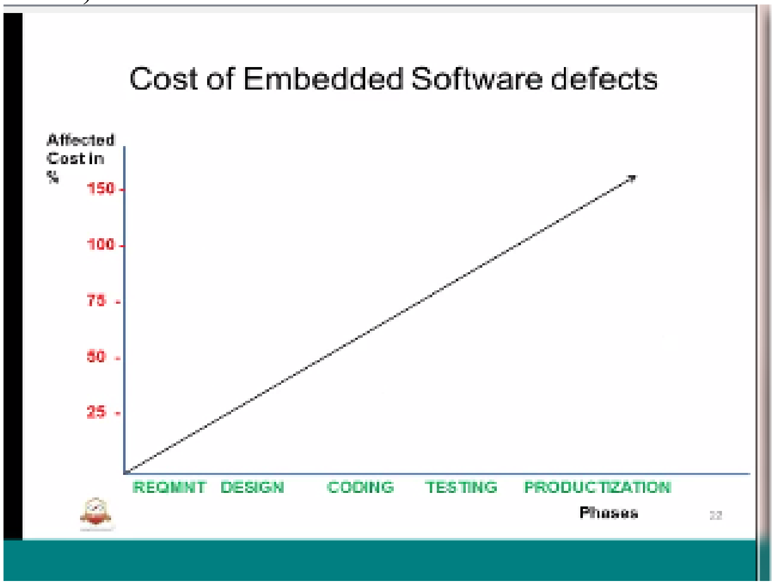


Why Testing

- Developers' oversight
- Certain bugs easier to find in testing
- Post release debugging is expensive
- Bugs in development tools such as compilers, library
- Best way to understand and present the product is to test it and report as PASS !
- Avoid customers detecting the defects
- Testing is challenging and rewarding

The why testing because some of the find in leafing post release debugging development tools such as complier, library. And we do not to compiler they complain for the customers identify derive bug. And we had analysis on cost of embedded software defects.

(Refer Slide Time: 22:22)

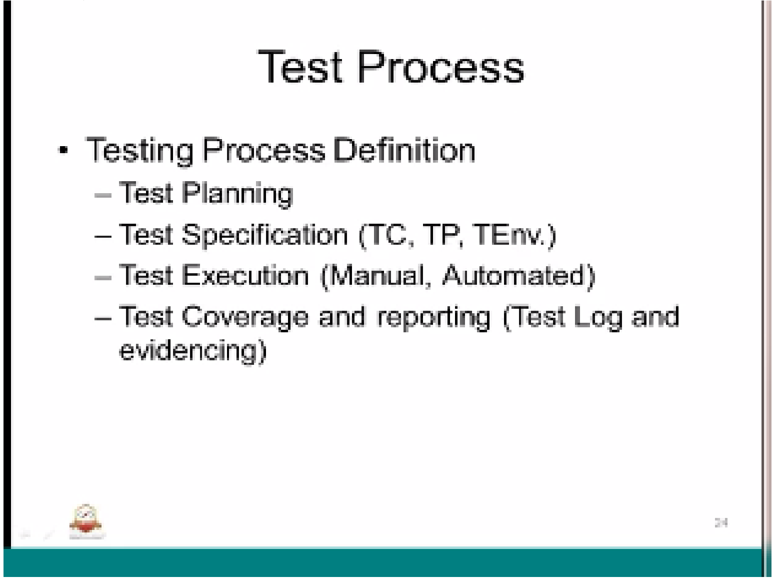


Cost of Embedded Software defects

Cost will increase we find defects Column and have we go along the different ways of the embedded software system? The cost will double there are few elements which are we monitory used for embedded software system for planning which talk about the plans of development and verification and contribution and technically inputs for embedded software are testing the activity.

The system requirement of the3 software requirement it standard design code then we have fixed results all together technical input or change the products for then we of course guild lines and standard the which are the used for redevelopment of the above all. We have testing process defined basically test planning.

(Refer Slide Time: 23:53)



Test Process

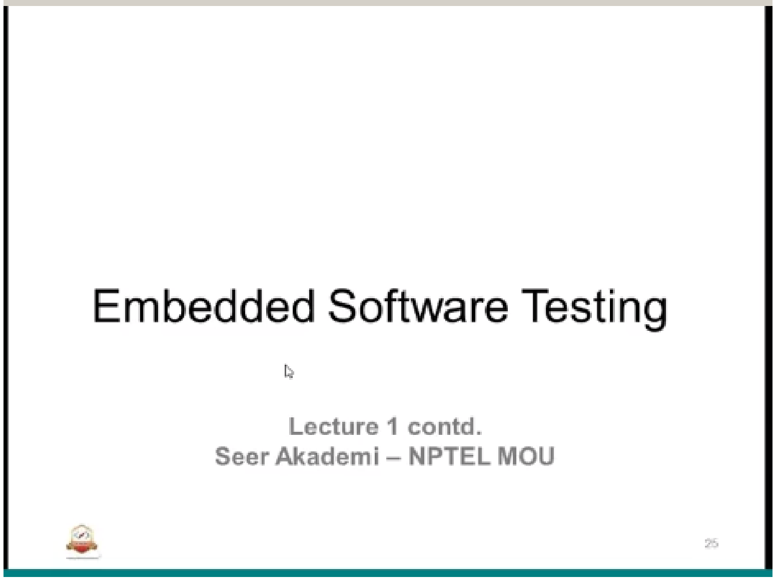
- Testing Process Definition
  - Test Planning
  - Test Specification (TC, TP, TEnv.)
  - Test Execution (Manual, Automated)
  - Test Coverage and reporting (Test Log and evidencing)

24

It talks about what is mean test plan I mean what should of at testing I am going to having process standard by imbrues, what are the causes I follow etc so lay out then we have the corresponding specification which will have the same procedure that the environment how let us dialogue that then what is the specification and test execution procedure environment is ready we do not have the execution.

This execution would be manually it bit automated. So then we have test coverage and then we have test specification and we will have a report, text report having the output agree test execution evidencing in the test log.

(Refer Slide Time: 24:55)

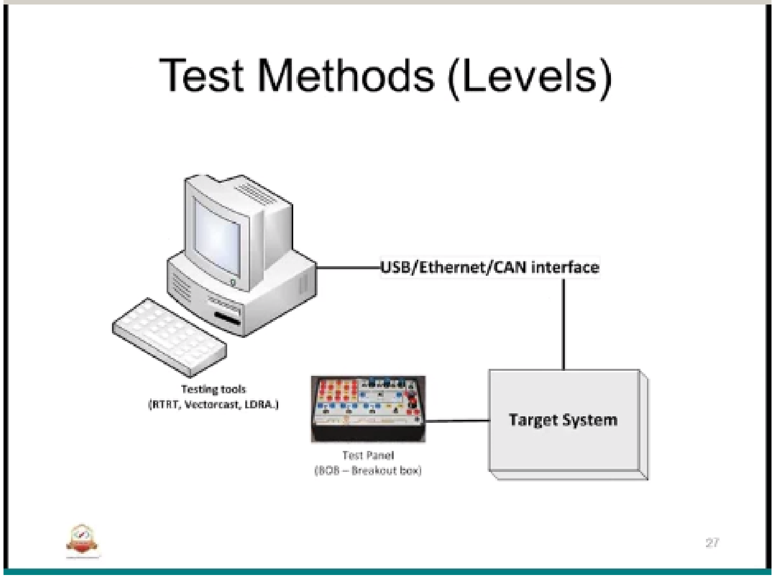


Embedded Software Testing

Lecture 1 contd.
Seer Akademi – NPTEL MOU

25

Now coming to embedded software testing the environment how the set up will be.
(Refer Slide Time: 25:07)



Test Methods (Levels)

USB/Ethernet/CAN interface

Testing tools
(RTRT, Vectorcast, LDRA.)

Test Panel
(BOB – Breakout box)

Target System

27

System testing setup will have comp lateral, again we have a laptop or desktop and we can interface with target system, target system meaning handmade system which is under test so we will have we, have an interface using an USB, or Ethernet, or CAN, so which will have the test the exhibition interface result whatever is the do you have this thing driven from the system desktop, and this thing that we have to enter in the system.

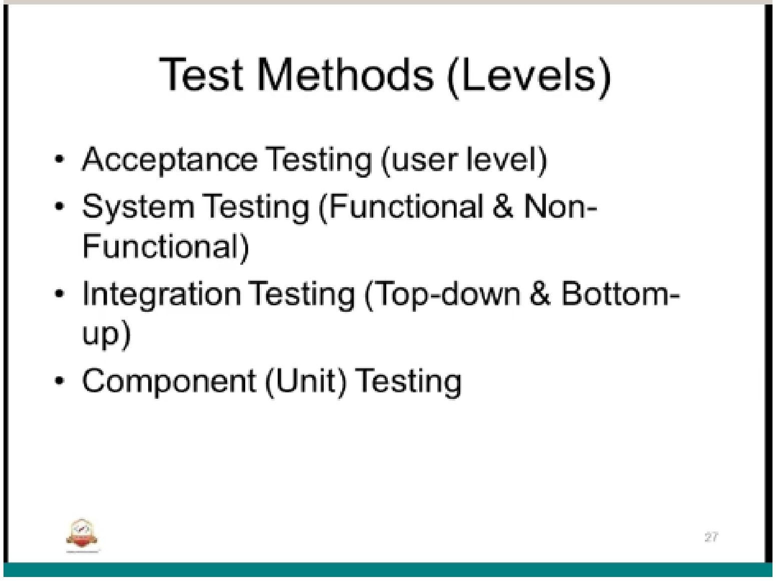
It will be done in the USB or and for the first given the system in the desktop the user level, so you will have the tools are a script, tools so MSE, RTRT, vector cast, IDRA these are basically know the tools, along with that we can also have typical mechanism which is basically the figures some of the input that are required for the specific system, the input figure the input data and what is the output.

The output also can be monitor are viewed to the same interface, USB function like that interface booked to the embedded system for the USB purposes, and the interface has been used as a input as well as the output okay, now coming to the embedded system having user features something like as a black box how can be trig the system like a said we can try the data from the desktop as well as there is a alternative mechanism.

To booked into the embedded system. This will be the nearest endorser nor mate, something it is called as a text panel also called as a breakout box, so what we do is see embedded system we will have a specific application turning on the time and certain data will not be realistic, so when it is running in the field only it will have realistic inputs but within the lab environment or embedded system is a thing setup, how are you going to read it, so there is a mechanism call up as a panel.

The text panel will have a typical desecrator inputs unlog inputs that has potential method etcetera, also there are test box useful some of the Ethernet messages I to see interface procedure, so to produce some of the realistic values we use the text panel, the same text panel data can be recoded or viewed as a result program less and uses that , so this will be the debugging or the system and this will be the text panel to the target system will be the most like the end to end process is called as end to end okay once we have.

(Refer Slide Time: 29:34)



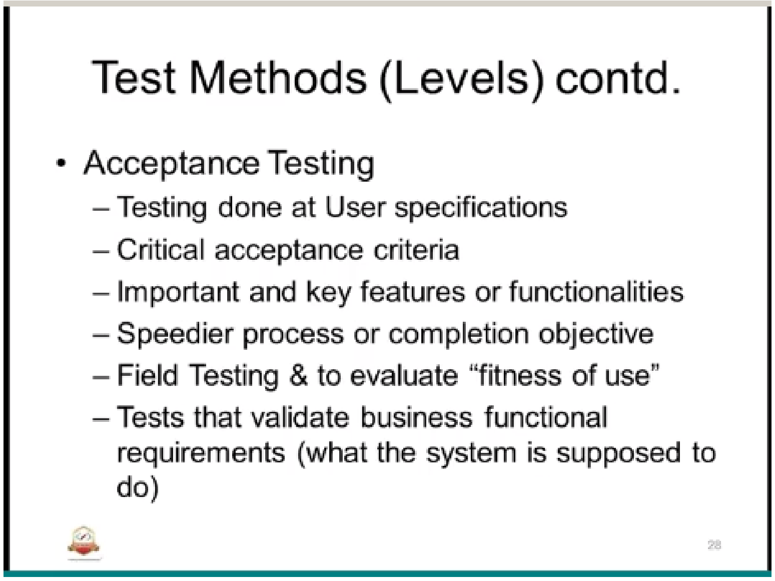
Test Methods (Levels)

- Acceptance Testing (user level)
- System Testing (Functional & Non-Functional)
- Integration Testing (Top-down & Bottom-up)
- Component (Unit) Testing

27

We know I mean we have seen how test method be approached how test method will be that, there are acceptance application system different test methods of the level there are used, the difficult one is in the end, the acceptance testing the user level acceptance testing system testing, functional and nonfunctional then we have integration testing and we have component testing, so there are four testing level one is acceptance testing, system testing, integration testing, component testing.

(Refer Slide Time: 30:19)



Test Methods (Levels) contd.

- Acceptance Testing
  - – Testing done at User specifications
  - – Critical acceptance criteria
  - – Important and key features or functionalities
  - – Speedier process or completion objective
  - – Field Testing & to evaluate “fitness of use”
  - – Tests that validate business functional requirements (what the system is supposed to do)

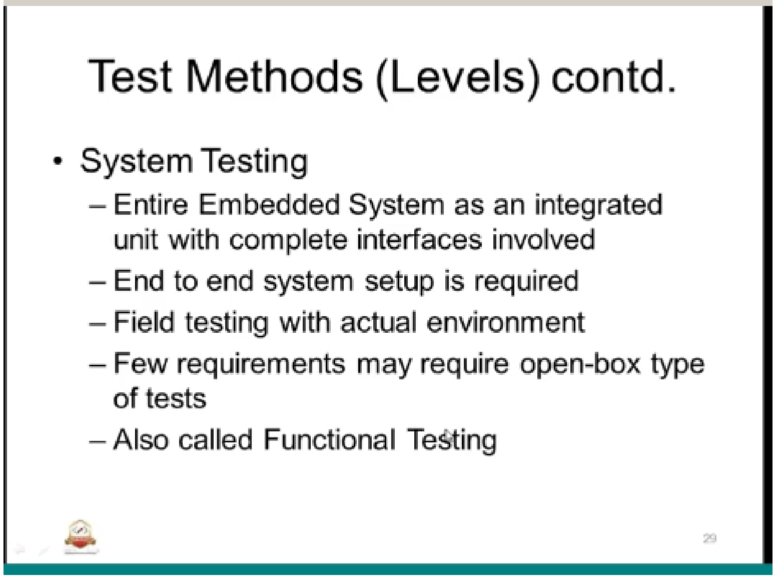
28

Each one we will acceptance testing or user system user acceptance is also called, these are based on the user specification that means you want the M brand system or the M product to use the how we will view it, that is what this testing we talked about, basically there are certain critical requirement for given acceptance criteria, which is very fast these are the thought of the acceptance testing those requirement.

Important key features of functionalities are suppose you take a mobile in this system, the important are what they specify in the requirement are specification I should have conference call I should have unable Wi- Fi it should have blue tooth with it, it should have internet, browsing that are available so photo, video, audio recording so many, those are all some of the user specification this will be tested under acceptance testing.

Really acceptance testing will not be a previous one minimum set of requirements will be part of this, completion objective will be there FIDDR is there, usually it is tested on the screen, with the help of the user or against the respected evaluate, the respected user or it can be used by the user and the accepting tests that validate business functional requirement what does it mean, what the system is suppose to do we should valid, these are some of the keep under process user acceptance that is the first level of testing then we have system testing which is the next level of system testing.

(Refer Slide Time: 32:50)



Test Methods (Levels) contd.

- System Testing
  - Entire Embedded System as an integrated unit with complete interfaces involved
  - End to end system setup is required
  - Field testing with actual environment
  - Few requirements may require open-box type of tests
  - Also called Functional Testing

29

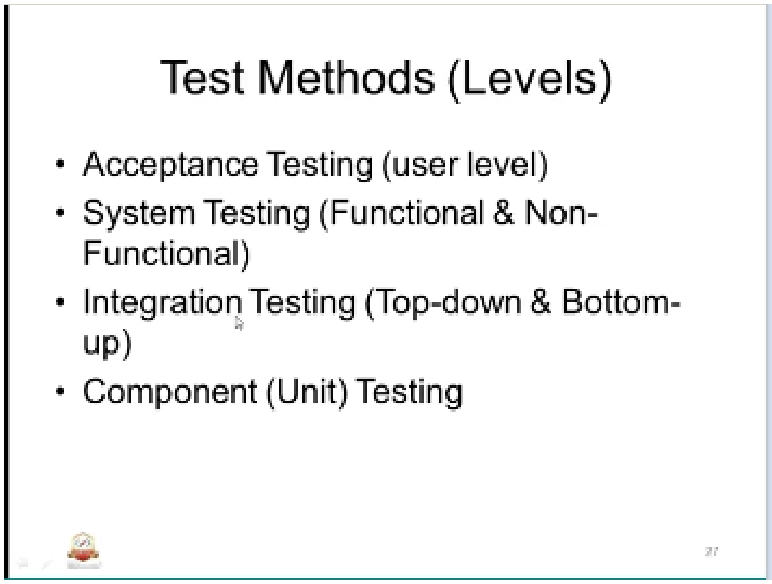
Entire embedded system testing we have the entire embedded system has integrated unit with complete interface involved, main in the entire end product will be established as a system, so other side the embedded system set up will be used and embedded system will be used for the interface the end to end system.

Will have all the features in the functionality and the interface that are required with the help of which this embedded system will be completely tested. So look at this what I was visioning the text panel and user are all electric inputs against the target system this will be the primary system testing mechanism, and the system testing will have an actual environment, where how the system is used in the field gracefully for example.

If you want to use the some of the components that are using in the car, vehicle will be using this card in the testing foots are the environmental foots coming from the card speed of the car, while driving the car, it may not be enough to have the actual environment in the field. So what will happen is some of the requirements.

We may have to break the environment system box that is called as a open box type of tests, for example some of the capability requirement index performances memory performance of many to be tested by opening the embedded system which may result in validating in some of the implemented code or going to the code debugging code open box testing the system is also called functional testing where all the functions will be tested so that is about the second level of testing.

(Refer Slide Time: 35:41)



Test Methods (Levels)

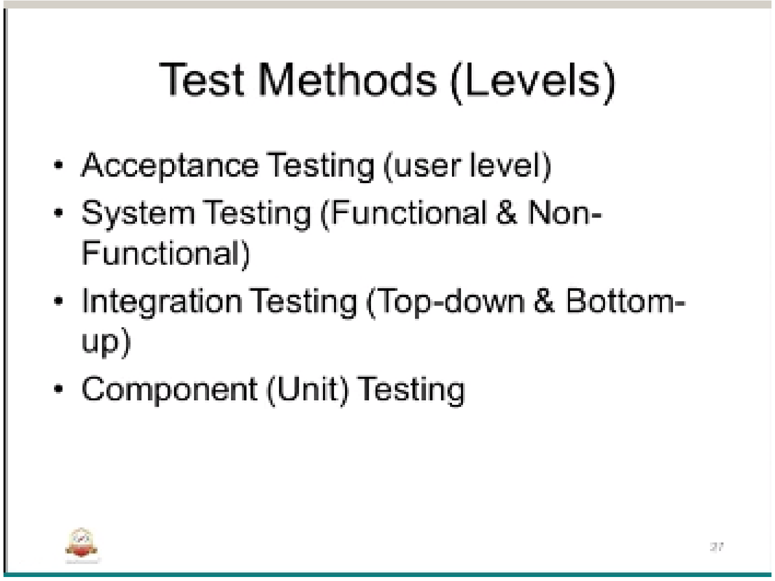
- Acceptance Testing (user level)
- System Testing (Functional & Non-Functional)
- Integration Testing (Top-down & Bottom-up)
- Component (Unit) Testing

We have the next method called integration testing what we do integration testing so basically integration testing we will have integrations of some modules of the system under test and which we evaluate the integration of the all these modules basically we will suppose 10 modules all this 10 modules they are called 10 modules, we will have to be interacted we will have evaluated again each modules how they interact.

How are the data some of the signals interacted in these modules all this part of the integration testing could be done at two approaches one is top-down approaches. top down integration where high level procedure or requirement are addressed and then low level from the requirements then two design then to code I will be addressing in order to complete the testing like the modules so higher level.

Next method of integration we do is bottom up why because top down approach may not be enough at the complete requirement testing so we may have to do some of the level integration that means the logical units only need to be we will begin with unit level functions possible units are first integrated then they area group into the next level etc. so higher level this all this units are tested so this is make it method of integration testing.

(Refer Slide Time: 37:57)



Test Methods (Levels)

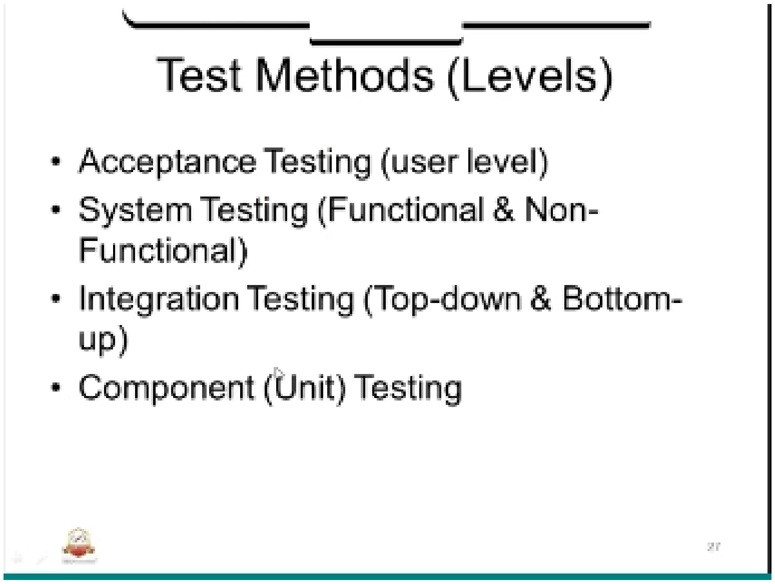
- Acceptance Testing (user level)
- System Testing (Functional & Non-Functional)
- Integration Testing (Top-down & Bottom-up)
- Component (Unit) Testing

We are seen acceptance a user level interesting system testing integration testing the last one is a the component level testing so what we do in a component level testing the smallest units functionality access of the embedded system will be testing other unit there are 10 modules each modules will have number of functionality that may have 1 2 10 tested so typically echo system procedure will be called as unit testing.

So typically a code function procedure, all these should be tested as a unit, what is the functionality it can address output ,also we have coverage, aspects that means we need to cover some different perspective, it is nothing but the structural coverage ,it is important to have a perspective coverage, that means 100% of the functionalities have to be covered, or it could be done with the help of statement, that it is implemented statement , or logical branch .The modified condition, or decision statement that is ,if else statement, for loops, this will be used as an coverage, there are tools, that are used, all will be used as a coverage, the tools that are used for testing.

Basically help in generate, which will generate an instrumented code ,provided the test code, that instrument code, is the run on the target, and the limit under that will be tested on the parse code criteria, here I said the logical units, are used, usually or in a group manner, so all the units once it is completed, it is called as unit testing, in a typical system where we have a complex functionality ,if they so it may not be able to possibly for us to completely test the one method, that is the brief level or integration level.

(Refer Slide Time: 40:55)



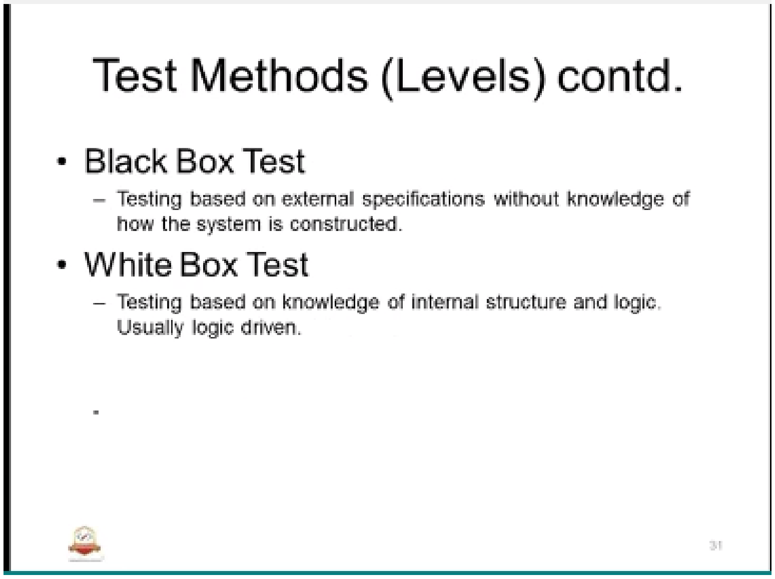
Test Methods (Levels)

- Acceptance Testing (user level)
- System Testing (Functional & Non-Functional)
- Integration Testing (Top-down & Bottom-up)
- Component (Unit) Testing

27

So we may have to do, credit based testing that means suppose unit testing will be able to test the coverage of 20% percentage, so that there is a 20% gap will contain the integration testing, the system testing, so testing they basically follow, to have a 100% coverage, and of course before deployment and fitness of this product, user acceptance testing is used basically, so these are some of the testing methods, that we have, used.

(Refer Slide Time: 41:46)



Test Methods (Levels) contd.

- Black Box Test
  - Testing based on external specifications without knowledge of how the system is constructed.
- White Box Test
  - Testing based on knowledge of internal structure and logic. Usually logic driven.

31

Now we have some categories of testing levels, we have the testing ,how it is broadly categorized, we have the test box available, so the test can be done in the two ways, one is the black box test, and the other one is white box test, so as I said the black box test will be based on external specification, without the knowledge of how the system is constructed, that is you do not have to bother about the individual units, or the integrated model especially in the embedded system unit.

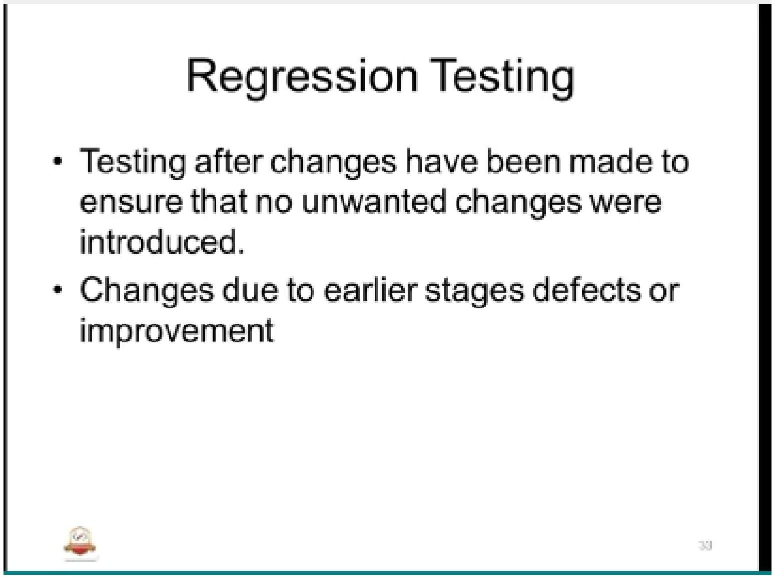
What do we have to, see how a black box how it can be triggered in the testing method, how we can expect the test output, typically as black box, for example a telephone instrument, you take it ,as a black box how are you going to test it ,is we connect it to the telephone wire, and telephone jack, then we will see whether the ring tone is coming, then we will dial some number, then we will ask redial, with the help of the buttons ,that r using the instrument, so we use as a black box, the complete telephone instrument.

This is black box testing. The next type of testing is the whit box testing, here we need to know the general logic, for this structure, how it is organized, so how it is designed, this is basically driven from the software design or the system design of the embedded system, but there are different groups logical groups of the embedded product modules, which are part of the embedded system, those will be tested, but usually something like we have 10 design module, which are responsible for 10 different modules.

As I said, when I take up a phone ,hooked with the telephone jack ,it should give a dial tone, the ring tone ,how it comes, will be responsible by the speed test or the summary function, those speakers will be separately, addressed in the white box test method, so usually it will be in the logical driven, it could be further development also , we can use it where some of the lowest possible requirements, of functionalities such as device driver or memory . We may not be able to see it at the black box.

What we are going to do is, we will use, the white box method, where we use the debugger, with the help of the debugger, you will monitor the memory, which will display the values, so we will see whether it is in the output, that is how it works with the intermediate, the next testing type is called regression testing.

(Refer Slide Time: 45:33)



Regression Testing

- Testing after changes have been made to ensure that no unwanted changes were introduced.
- Changes due to earlier stages defects or improvement

33

So what do, we do in a regression testing ,basically when we have made testing method ,which may result in, what will happen is this testing method will end in failure, or some changes may happen ,in terms of fixing the failure, so what will happen is , it is to be retested, so how it will be re tested is ,using regression testing, that means given a black box or white box , embedded system, we will basically address those tests ,which are failed.

Because those tests, which are failed earlier ,are supposed to pass now, because those failures are reported, and those failures have been fixed, so all we need to do, is re execute, we will not change our approach, or we will not modify the test, what we will do is, we will simply run the embedded system, which is having the fixed system of the software, we use a method called regression testing ,after changes have been made to ensure that no unwanted changes were introduced, that means according ,to the features .

What are required ,there shall be an additional data ,the changes due to earlier stages ,defects or improvement , the changes need not be relatively in any of the requirement , and the changes could be on the failures, of the earlier stages and that is all about the testing and the test methods, and the type of testing used in the test method and in the next session we will discuss the test case design and procedures.