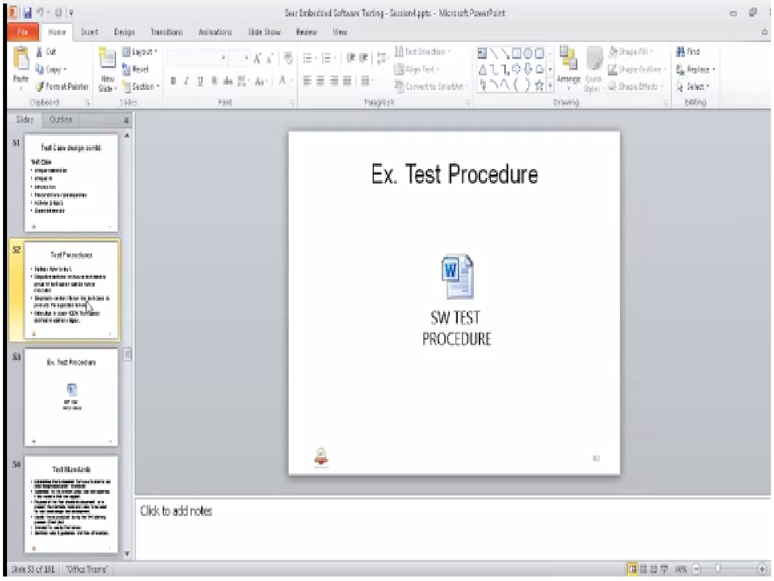
(Refer Slide Time: 00:06)



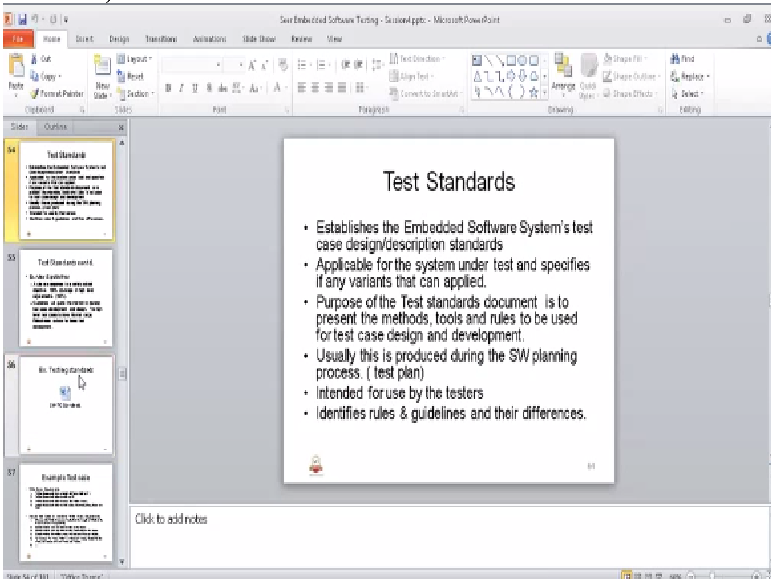
Ex. Test Procedure

SW TEST
PROCEDURE

Click to add notes

We will go through some of the test standards that are told in the earlier session,

(Refer Slide Time: 00:14)



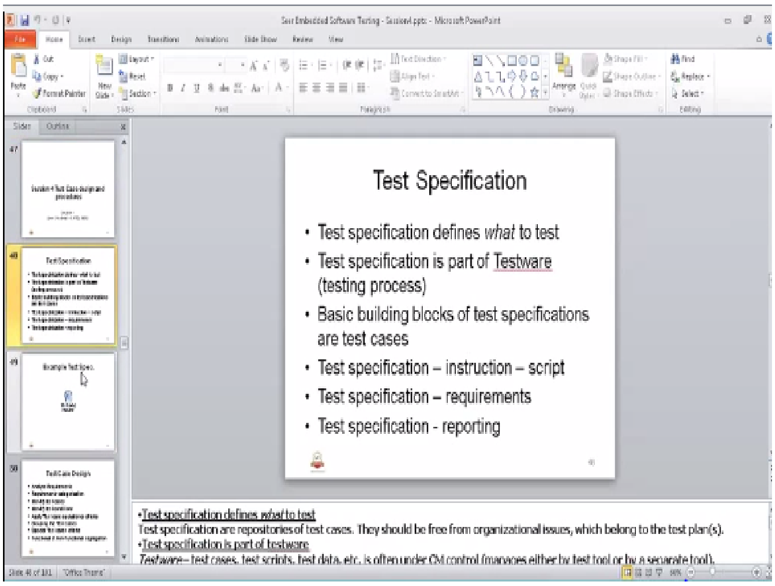
Test Standards

- Establishes the Embedded Software System's test case design/description standards
- Applicable for the system under test and specifies if any variants that can applied.
- Purpose of the Test standards document is to present the methods, tools and rules to be used for test case design and development.
- Usually this is produced during the SW planning process. (test plan)
- Intended for use by the testers
- Identifies rules & guidelines and their differences.

Click to add notes

Like test standards, in order to have the test plan, specification we will go through only the test standard with an example and recap some of the session inputs from some of the last class so,

(Refer Slide Time: 00:46)

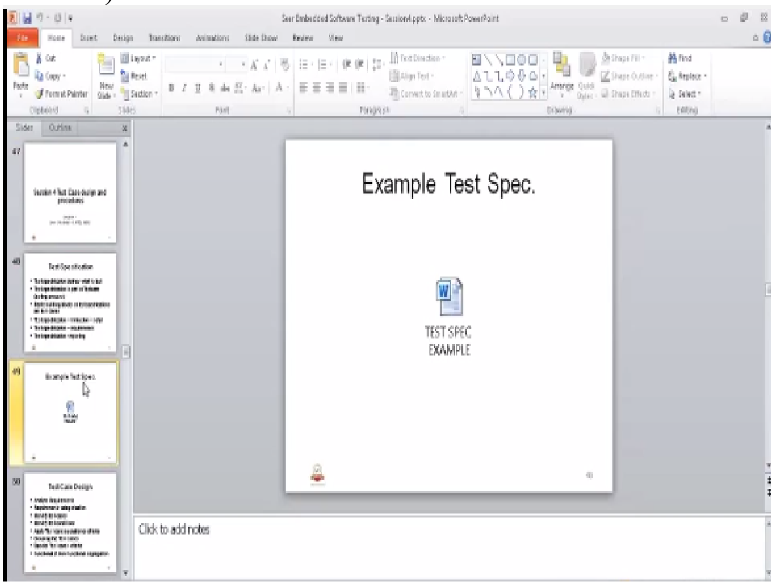


Test Specification

- 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

•Test specification defines *what* to test
Test specification are repositories of test cases. They should be free from organizational issues, which belong to the test plan(s).
•Test specification is part of testware
Testware – test cases, test scripts, test data, etc. is often under CM control (manages either by test tool or by a separate tool).

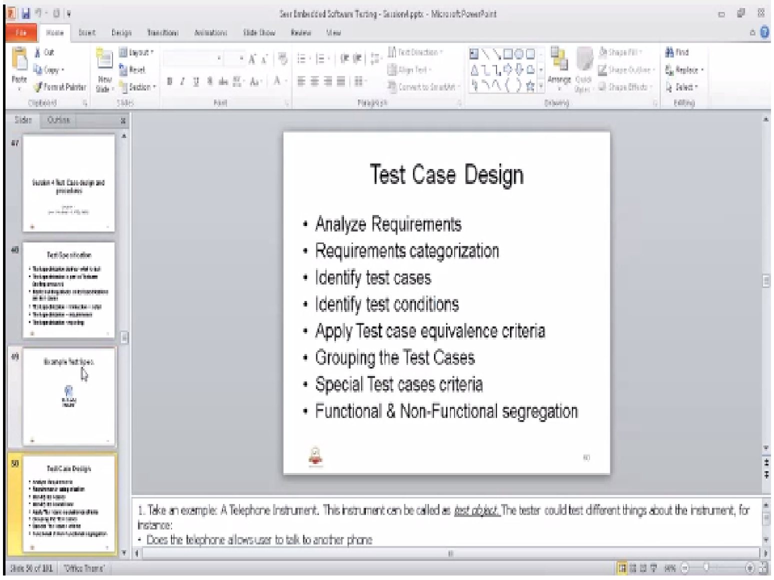
We will go through the specification example test spec
(Refer Slide Time: 00:48)



Example Test Spec.

Click to add notes

How to write and what is the format that is contains of test spec and then
(Refer Slide Time: 00:57)

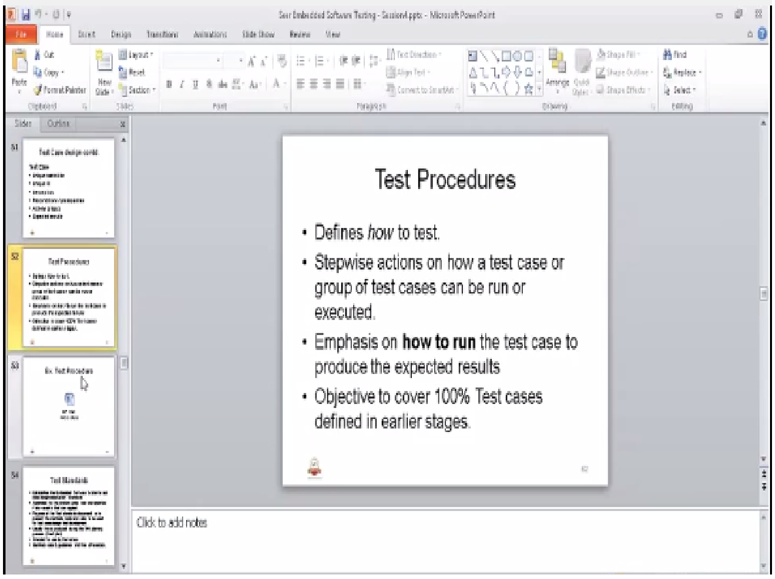


Test Case Design

- 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

1. Take an example: A Telephone Instrument. This instrument can be called as *test object*. The tester could test different things about the instrument, for instance:
- Does the telephone allows user to talk to another phone

How it is design. So what are the steps that is needed similarly, test procedures also
(Refer Slide Time: 01:16)

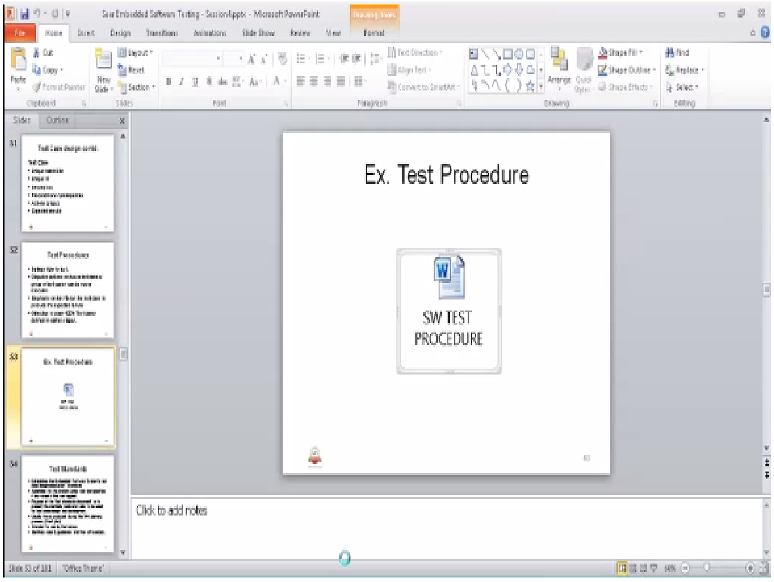


Test Procedures

- Defines *how* to test.
- Stepwise actions on how a test case or group of test cases can be run or executed.
- Emphasis on **how to run** the test case to produce the expected results
- Objective to cover 100% Test cases defined in earlier stages.

Click to add notes

We have discussed the test procedure should be 100% covering to the test case and
(Refer Slide Time: 01:16)

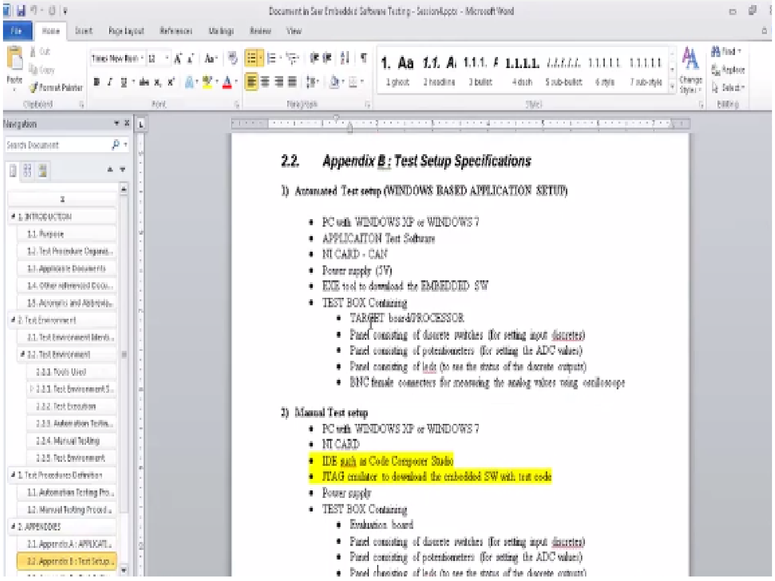


Ex. Test Procedure

SW TEST PROCEDURE

Click to add notes

We went through the test procedure example which identify the grouping of the test table this we have already identified in the test specification and that is grouping will be done that procedure is what we do is practical steps of the test table that test procedure will be independent which have mention in the test environment in each of the specification.
We will have the specification and we will identified various tools and that are test based whatever it is one is automated and other one is manual like both are similar in terms of the tools and other set of excepted we use it worked approach
(Refer Slide Time: 02:30)



2.2. *Appendix B : Test Setup Specifications*

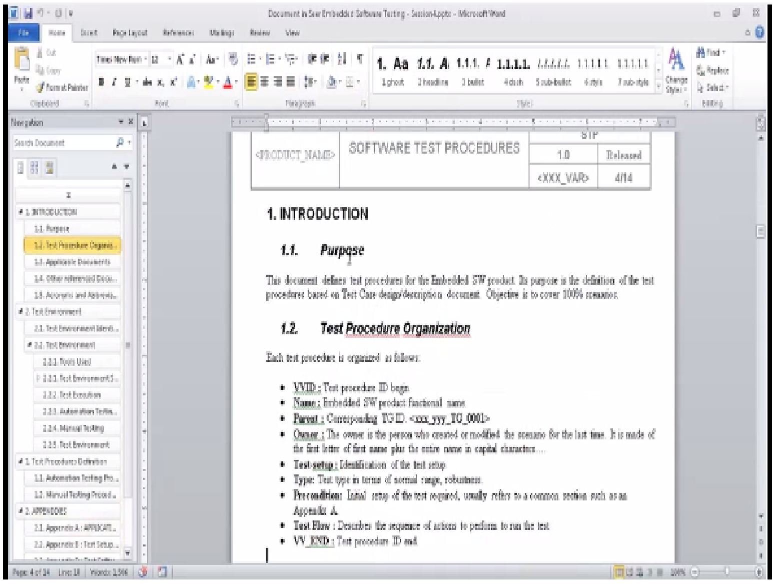
1) Automated Test setup (WINDOWS BASED APPLICATION SETUP)

- PC with WINDOWS XP or WINDOWS 7
- APPLICAITON Test Software
- NI CARD - CAN
- Power supply (5V)
- EXE tool to download the EMBEDDED SW
- TEST BOX Containing
  - TARGET board/PROCESSOR
  - Panel consisting of discrete switches (for setting input discretes)
  - Panel consisting of potentiometers (for setting the ADC values)
  - Panel consisting of leds (to see the status of the discrete outputs)
  - BNC female connectors for measuring the analog values using oscilloscope

2) Manual Test setup

- PC with WINDOWS XP or WINDOWS 7
- NI CARD
- IDE such as Code Composer Studio
- JTAG emulator to download the embedded SW with test code
- Power supply
- TEST BOX Containing
  - Evaluation board
  - Panel consisting of discrete switches (for setting input discretes)
  - Panel consisting of potentiometers (for setting the ADC values)
  - Panel consisting of leds (to see the status of the discrete outputs)

Procedure in earlier session we went through the example for this procedure especially organisation
(Refer Slide Time: 02:43)


Document in Saw Embedded Software Testing - SessionAppts - Microsoft Word
Navigation
1. INTRODUCTION
1.1. Purpose
1.2. Test Procedure Organiz...
1.3. Applicable Documents
1.4. Other referenced Docu...
1.5. Acronyms and Abbrevia...
2. Test Environment
2.1. Test Environment Identi...
2.2. Test Environment
2.2.1. Tools Used
2.2.2. Test Execution
2.2.3. Automation Testin...
2.2.4. Manual Testing
2.2.5. Test Environment
1. Test Procedures Definition
1.1. Automation Testing Pro...
1.2. Manual Testing Proced...
2. APPENDIXES
2.1. Appendix A : APPLICATI...
2.2. Appendix B : Test Setup...

<PRODUCT_NAME>	SOFTWARE TEST PROCEDURES	STP	
		1.0	Released
		<XXX_VAR>	4/14

1. INTRODUCTION

1.1. Purpose

This document defines test procedures for the Embedded SW product. Its purpose is the definition of the test procedures based on Test Case design/description document. Objective is to cover 100% scenarios.

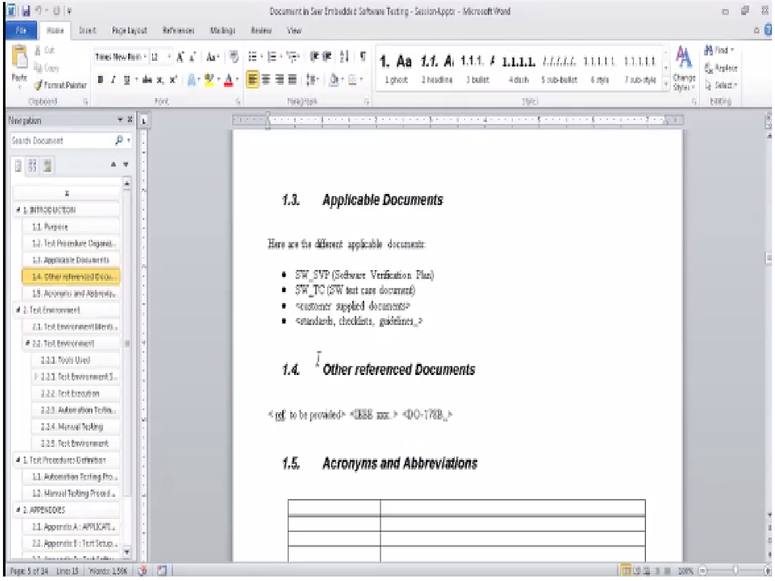
1.2. Test Procedure Organization

Each test procedure is organized as follows:

- VVID : Test procedure ID begin
- Name : Embedded SW product functional name
- Parent : Corresponding TG ID: <xxx_yyy_TG_0001>
- Owner : The owner is the person who created or modified the scenario for the last time. It is made of the first letter of first name plus the entire name in capital characters....
- Test-setup : Identification of the test setup
- Type: Test type in terms of normal range, robustness.
- Precondition: Initial setup of the test required, usually refers to a common section such as an Appendix A.
- Test Flow : Describes the sequence of actions to perform to run the test
- VV_END : Test procedure ID end

Which we gone through all the test cases with that we are run to that and,

(Refer Slide Time: 02:55)


Document in Saw Embedded Software Testing - SessionAppts - Microsoft Word
Navigation
1. INTRODUCTION
1.1. Purpose
1.2. Test Procedure Organiz...
1.3. Applicable Documents
1.4. Other referenced Docu...
1.5. Acronyms and Abbrevia...
2. Test Environment

1.3. Applicable Documents

Here are the different applicable documents:

- SW_SVP (Software Verification Plan)
- SW_TC (SW test case document)
- <customer supplied documents>
- <standards, checklists, guidelines_>

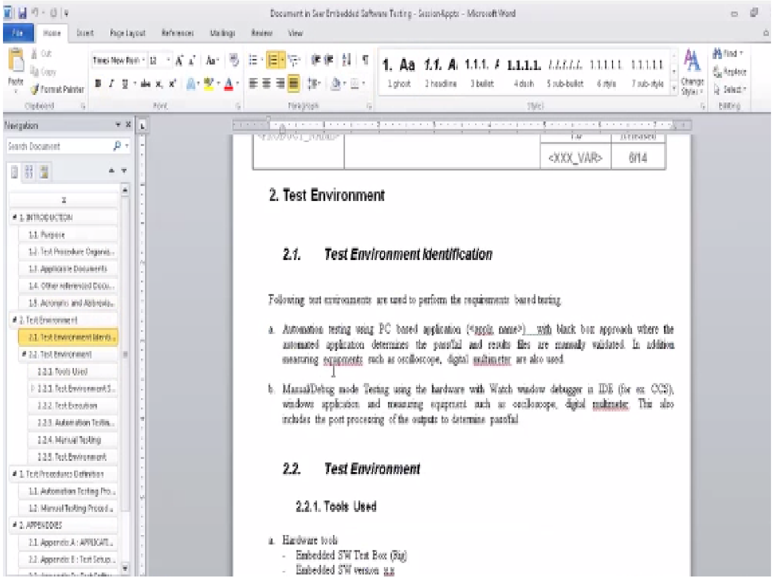
1.4. Other referenced Documents

< ref to be provided> <IEEE xxx_> <DO-178B_>

1.5. Acronyms and Abbreviations

We will refer some of the external document and internal document system of separate documents also.

(Refer Slide Time: 03:05)



<PRODUCT_NAME>		1.0	Released
		<XXX_VAR>	8/14

2. Test Environment

2.1. *Test Environment Identification*

Following test environments are used to perform the requirements based testing

a. Automation testing using PC based application (<appl. name>) with black box approach where the automated application determines the pass/fail and results files are manually validated. In addition measuring equipments such as oscilloscope, digital multimeter are also used

b. Manual/Debug mode Testing using the hardware with Watch window debugger in IDE (for ex. CCS), windows application and measuring equipment such as oscilloscope, digital multimeter. This also includes the post processing of the outputs to determine pass/fail

2.2. *Test Environment*

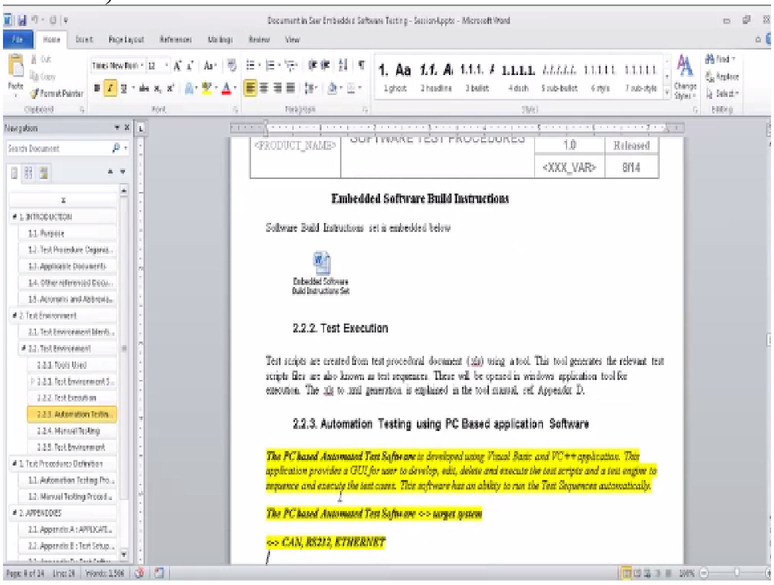
2.2.1. Tools Used

a. Hardware tools
- Embedded SW Test Box (Rig)
- Embedded SW version x.x

We had discussed automation testing a set based of equation and manual testing for the user debugger that is the user has re controlled that of the program coming on the embedded surface also we are reading the tools used the hardware tools and software tools I will go throw all this and explain this conclusion and embedded test if you have any questions you can ask to me on this example okay.

Then we had discussed about application set up procedure how it should be lay out and to build a program of the embedded software how we are going to do it etc.

(Refer Slide Time: 04:14)



<PRODUCT_NAME>	SOFTWARE TEST PROCEDURES	1.0	Released
		<XXX_VAR>	8/14

Embedded Software Build Instructions

Software Build Instructions set is embedded below

Embedded Software Build Instructions Set

2.2.2. Test Execution

Test scripts are created from test procedural document (xls) using a tool. This tool generates the relevant test scripts files are also known as test sequences. These will be opened in windows application tool for execution. The xls to xml generation is explained in the tool manual, ref. Appendix D.

2.2.3. Automation Testing using PC Based application Software

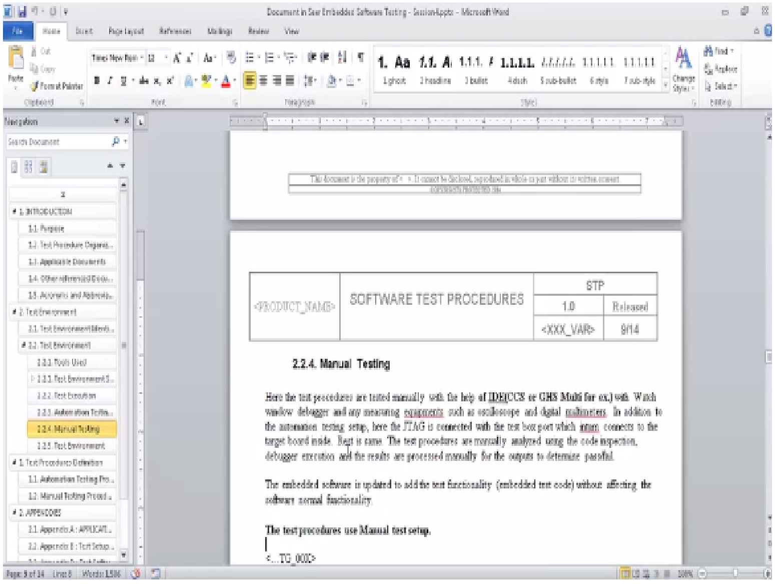
The PC based Automated Test Software is developed using Visual Basic and VC++ application. This application provides a GUI for user to develop, edit, delete and execute the test scripts and a test engine to sequence and execute the test cases. This software has an ability to run the Test Sequences automatically.

The PC based Automated Test Software <-> target system

<-> CAN, RS232, ETHERNET

Also we came to know that test execution can be done throw teat base are developed based on the test it can be part of a microchip, microchip can be used for automating the easy testing

(Refer Slide Time: 04:36)



Document in Seer Embedded Software Testing - SessionAppts - Microsoft Word
Navigation
Search Document

This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

<PRODUCT_NAME>	SOFTWARE TEST PROCEDURES	STP	
		1.0	Released
		<XXX_VAR>	9/14

2.2.4. Manual Testing

Here the test procedures are tested manually with the help of IDE(CCS or GHS Multi for ex.) with Watch window debugger and any measuring equipments such as oscilloscope and digital multimeters. In addition to the automation testing setup, here the JTAG is connected with the test box port which intern connects to the target board inside. Rest is same. The test procedures are manually analyzed using the code inspection, debugger execution and the results are processed manually for the outputs to determine pass/fail.

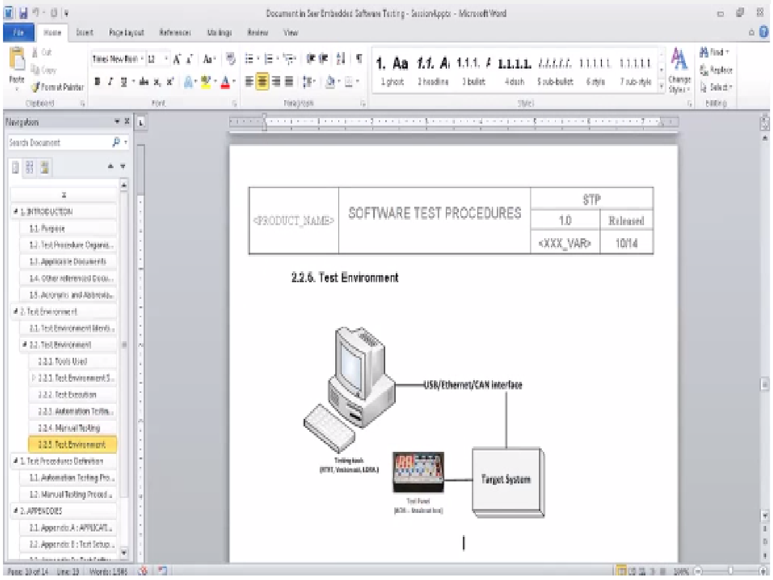
The embedded software is updated to add the test functionality (embedded test code) without affecting the software normal functionality.

The test procedures use Manual test setup.

<...TG_00X>

Then we start discussing about the manual testing which has integrated development environment to tell the multi data varies separate procurer and this are steps for manual testing.

(Refer Slide Time: 04:53)



Document in Seer Embedded Software Testing - SessionAppts - Microsoft Word
Navigation
Search Document
USB/Ethernet/CAN Interface
Testing tools
Target System
Test Panel

<PRODUCT_NAME>	SOFTWARE TEST PROCEDURES	STP	
		1.0	Released
		<XXX_VAR>	10/14

2.2.6. Test Environment

How you do the typical test environment for the test case so where in the tools all that interfaces had given for the PC the interface could be connected with the target system and to feed the really time values we use a test panel witch will be in the test screens, then we stated aggregate automated test procedure test flow is written here,

(Refer Slide Time: 05:34)

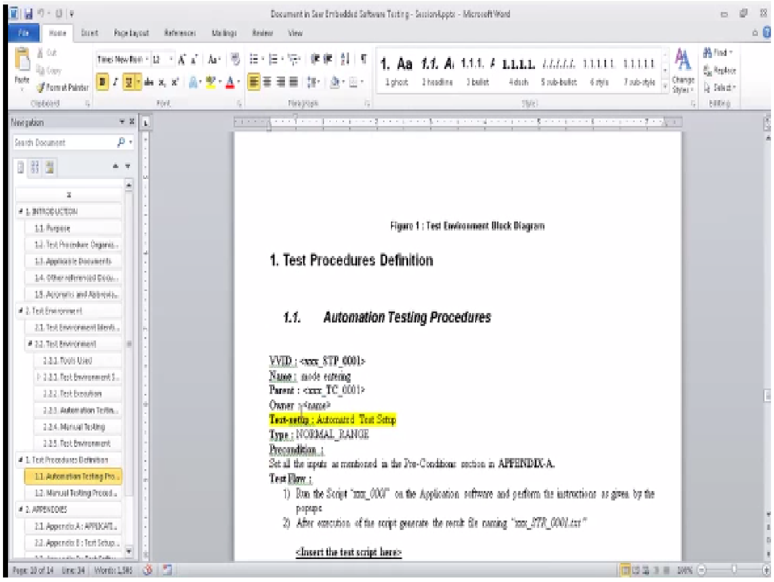


Figure 1 : Test Environment Block Diagram

1. Test Procedures Definition

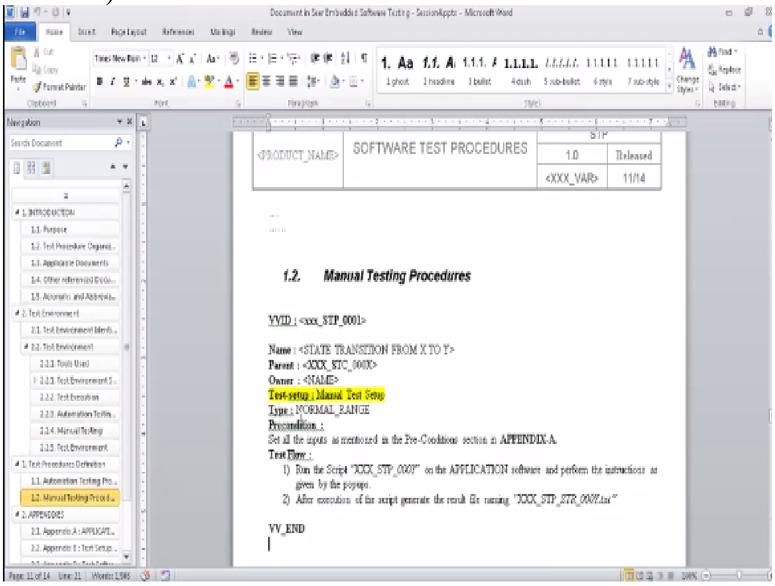
1.1. Automation Testing Procedures

VVID : <xxx_STP_0001>
Name : mode entering
Parent : <xxx_TC_0001>
Owner :<name>
Test-setup : Automated Test Setup
Type : NORMAL_RANGE
Precondition :
Set all the inputs as mentioned in the Pre-Conditions section in APPENDIX-A.
Test Flow :
1) Run the Script "xxx_0001" on the Application software and perform the instructions as given by the popups
2) After execution of the script generate the result file naming "xxx_STR_0001.txt"

<Insert the test script here>

And the script what draft in the earlier pages it has been inserted or it can refer to the configure scripts this only the manual processing this also can be done throw script the script also can developed for an IDE.

(Refer Slide Time: 05:55)



<PRODUCT_NAME>	SOFTWARE TEST PROCEDURES	STP	
		1.0	Released
		<XXX_VAR>	11/14

....
......

1.2. Manual Testing Procedures

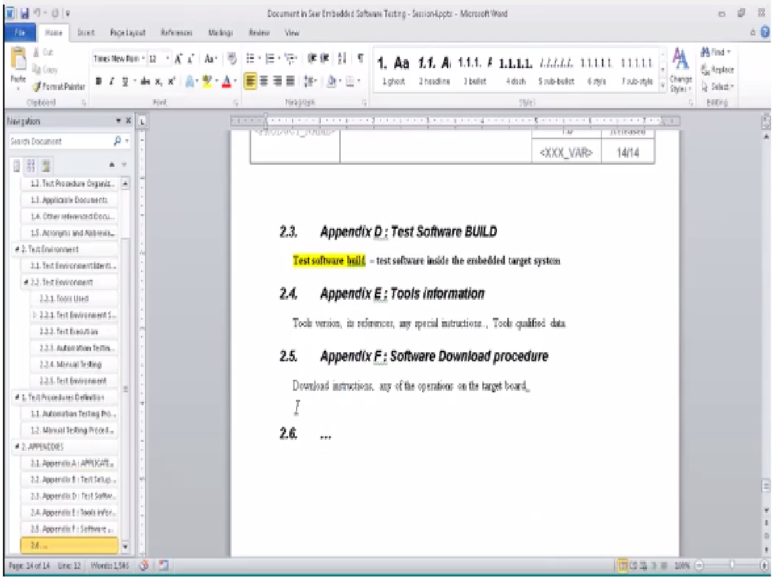
VVID : <xxx_STP_0001>

Name : <STATE TRANSITION FROM X TO Y>
Parent : <XXX_STC_000X>
Owner : <NAME>
Test-setup : Manual Test Setup
Type : NORMAL_RANGE
Precondition :
Set all the inputs as mentioned in the Pre-Conditions section in APPENDIX-A.
Test Flow :
1) Run the Script "XXX_STP_000Y" on the APPLICATION software and perform the instructions as given by the popups.
2) After execution of the script generate the result file naming "XXX_STP_STR_000Y.txt"

VV_END

To integration the development environment which will have conisation in the Debugger, debugger access command there we can have a P conditions are well be handing manually used before we started that is it could 13 values or it could be whatever it is,

(Refer Slide Time: 06:44)



		<XXX_VAR>	14/14

2.3. Appendix D : Test Software BUILD

Test software build – test software inside the embedded target system

2.4. Appendix E : Tools information

Tools version, its references, any special instructions., Tools qualified data

2.5. Appendix F : Software Download procedure

Download instructions, any of the operations on the target board.

2.6. ...

So also we can have information to detect the software, the software is something like personal build, we allow the user to monitor or to receive the data from the target term, then we will highlight the base operation of tools, and it has the certain instructions related to this system, it has to qualified or the data all in the software system, in this system appendix software download the procedure, it has to download the test procedure of system okay,

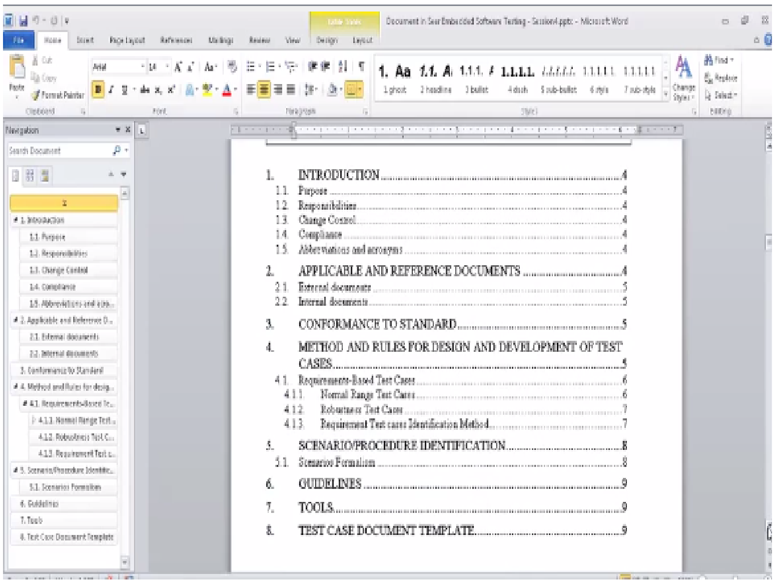
So we come to test standards for us to do all this actives test case, there rest is standard against which the test software are developed also it has other information interims of which will be used for validating, analysing the test execution base so test standards it establishes the embedded software systems test case design description standards applicable for the system under test and specifies.

If any variants that can applied that means it can changes for specific changes can be done all this standards will be highlighted and it highlights propos of the test standards in terms of methods tools that has to be used for test development so this standards well be in the test plan means the standards have to available or standards with the test plan.

For example a rule is a response to a critical objective 100 percentage coverage of high level requirements that means this rule can be the tester has to cover the requirements guidelines will guide the method to develop test case development and design the high level test cases to have normal range, robustness criteria for basic test development so all this design of all test cases in terms test improvement all this well be part of the test standards so we had gone throw an example of test standards

Basic template will be same as the other test cases, test planning where in all the equalised the full diagram tools and the revision all this well be kind of a trop.

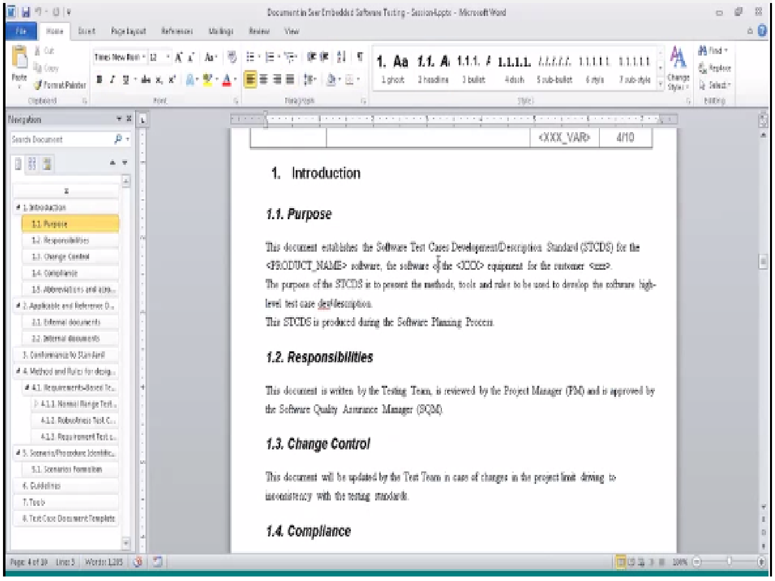
(Refer Slide Time: 10:15)



So basically test standards will have introduction like purpose, responsibilities change control will be identify in this session then we explained standard specification to ISO need to referred we will use this mentioned here so methods of rules designing the compact test cases for recalling those test cases I have put a example for targeting so is required for the test cases so that will identify in normal code that is the identification method.

Tell me how to identify the scenarios test case all this in the standard so first we have a guideline session, tools, test documents template the test template has to be pointed in the system so let us go throw the details of the test cases.

(Refer Slide Time: 11:45)



<XXX_VAR> 4/10

1. Introduction

1.1. Purpose

This document establishes the Software Test Cases Development/Description Standard (STCDS) for the <PRODUCT_NAME> software, the software of the <XXX> equipment for the customer <zzz>.
The purpose of the STCDS is to present the methods, tools and rules to be used to develop the software high-level test case dev/description.
This STCDS is produced during the Software Planning Process.

1.2. Responsibilities

This document is written by the Testing Team, is reviewed by the Project Manager (PM) and is approved by the Software Quality Assurance Manager (SQM).

1.3. Change Control

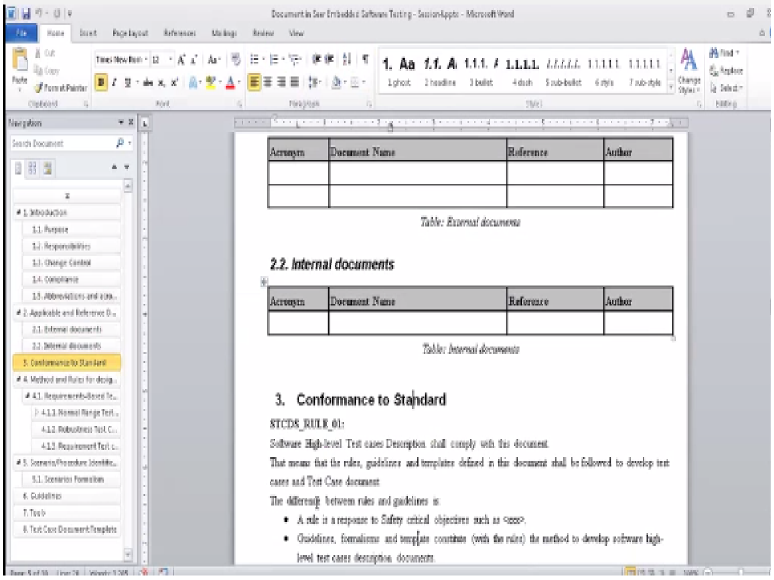
This document will be updated by the Test Team in case of changes in the project limit driving to inconsistency with the testing standards.

1.4. Compliance

So purpose we need to that means the document establish the software test cases development deception standard for the particular embedded software and temporary is going to that of so all its information and then what are the aspects of development and in test planning. This responsibilities tool developer uses is document is written by the testing and then it is received by the project manager (PM).

And is approved by the software quality assurance manager (SQM), Change control will be used for developing the test so, this document will be updated by the test tears in care of changes in the project limit driving to inconsistency with the testing standards. There are any

changes to be incorporated all that have to be controlled, compliance is internal components that you want to write there something like some templates, testing has to be compiled to that and then the applicable and reference documents which define and here the separate section with conformance to standard (DO178B, ISO26262).
(Refer Slide Time: 13:31)


Document in Seer Embedded Software Testing - Session-Apptx - Microsoft Word
File
Home
Insert
Page Layout
References
Mailings
Review
View
Navigation
Search Document
1. Introduction
1.1. Purpose
1.2. Responsibilities
1.3. Change Control
1.4. Compliance
1.5. Abbreviations and acro...
2. Applicable and Reference D...
2.1. External documents
2.2. Internal documents
3. Conformance to Standard
4. Method and Rules for desig...
4.1. Requirements-Based Te...
4.1.1. Normal Range Test...
4.1.2. Robustness Test C...
4.1.3. Requirement Test c...
5. Scenario/Procedure Identific...
5.1. Scenarios Formalism
6. Guidelines
7. Tools
8. Test Case Document Template

Acronym	Document Name	Reference	Author

Table: External documents

2.2. Internal documents

Acronym	Document Name	Reference	Author

Table: Internal documents

3. Conformance to Standard

STCDS_RULE_01:

Software High-level Test cases Description shall comply with this document.

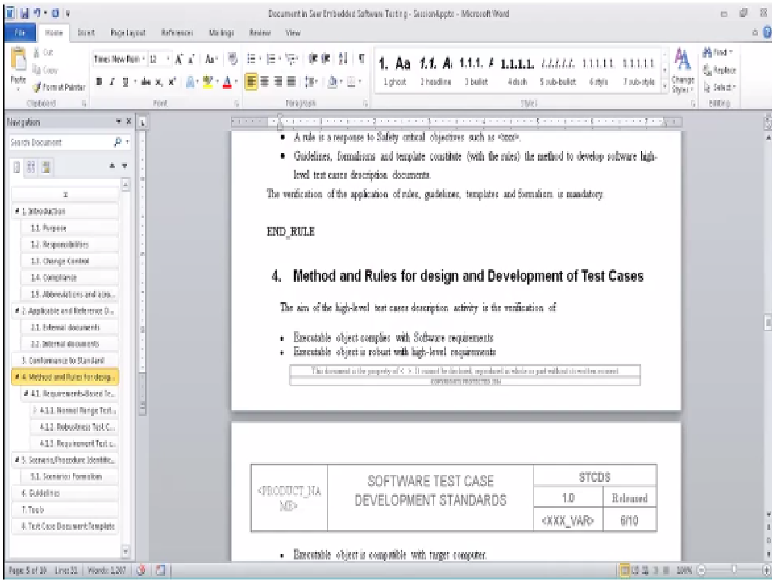
That means that the rules, guidelines and templates defined in this document shall be followed to develop test cases and Test Case document.

The difference between rules and guidelines is:

- A rule is a response to Safety critical objectives such as <xxx>.
- Guidelines, formalisms and template constitute (with the rules) the method to develop software high-level test cases description documents.

Next we come to rules, software high level test description shall compile this document, that means as per this document point outs the high level test basis, that the rules, guide lines and templates in this document define an test base on this document and this document talks about verifying diagrams and what are the differences, the rule is the responsible very critical of this one.

Guide lines is formal and the template comes to it all this will be sorted out, it is a part of the detainment that is the guide lines give a point to the user to understand the methods and what are the details about how test data should be developed, it is strictly a wide length that will be a evaluating style, the verification of the application of rules, guidelines, competence and formulate the mandatory, the test cases or procedures is done that needs to be verified, that is what the tool says this is itself one rules, now coming to methods and rules for design and development of the test cases,
(Refer Slide Time: 15:37)



- A rule is a response to Safety critical objectives such as <xxx>.
- Guidelines, formalisms and template constitute (with the rules) the method to develop software high-level test cases description documents.

The verification of the application of rules, guidelines, templates and formalisms is mandatory.

END_RULE

4. Method and Rules for design and Development of Test Cases

The aim of the high-level test cases description activity is the verification of

- Executable object complies with Software requirements
- Executable object is robust with high-level requirements

This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	6/10

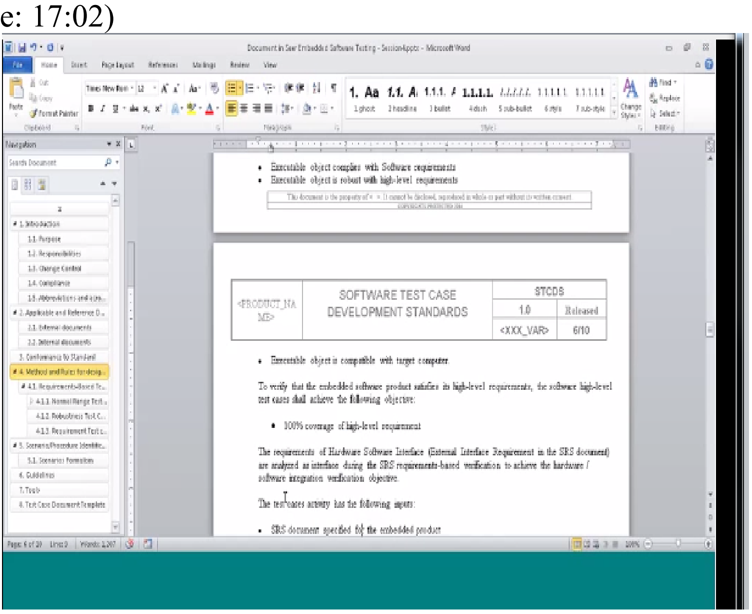
- Executable object is compatible with target computer.

The aim of the test cases to be is the verification of executable object which complies with the software requirement executable object is robust with high level requirements that mean executable program or the program that is embedded on the rpm so leave compliance with the software requirements should software to the SRS and it should be robust with the high level requirements.

And you should take care of all the robust spaces in terms of the test environmental development test cases. Based on to the software is compatible to the target competence, so the next one is verify that the embedded software product reaches its high level requirement the software high level test cases select the following objectives.

That means all the test cases that is developed and these are all few examples this can be tailored or it can be specific.

(Refer Slide Time: 17:02)



- Executable object complies with Software requirements
- Executable object is robust with high-level requirements

This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	6/10

- Executable object is compatible with target computer.

To verify that the embedded software product satisfies its high-level requirements, the software high-level test cases shall achieve the following objective:

- 100% coverage of high-level requirement

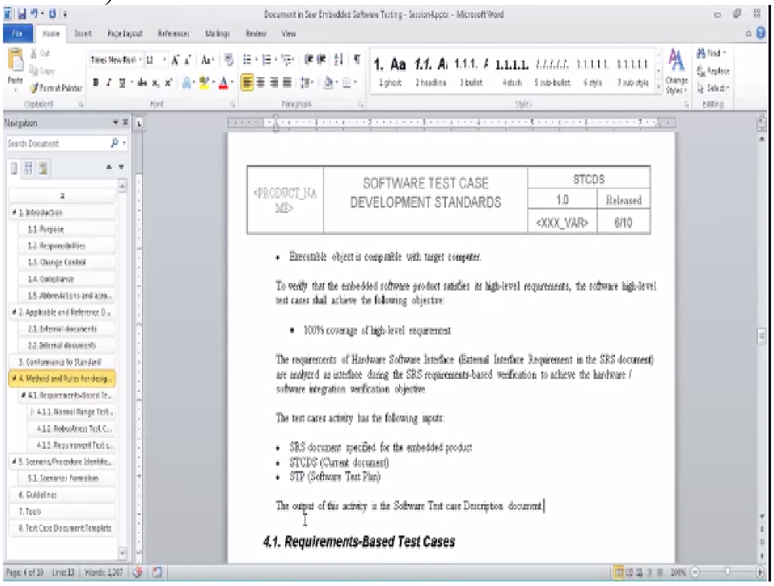
The requirements of Hardware Software Interface (External Interface Requirement in the SRS document) are analyzed as interface during the SRS requirements-based verification to achieve the hardware / software integration verification objective.

The test cases activity has the following inputs:

- SRS document specified for the embedded product

The requirements of hardware, software interfaces, these are the interface requirements in the SRS, it should analyse an interface during the SRS requirements based interfaces to achieve the hardware software integration software objectives. That is if someone wants to identify the integration the interfaces aspects of the SRS. The test cases activity following the SRS of these standards aspects this is a plan.

(Refer Slide Time: 17:44)



<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	6/10

- Executable object is compatible with target computer.

To verify that the embedded software product satisfies its high-level requirements, the software high-level test cases shall achieve the following objective:

- 100% coverage of high-level requirement

The requirements of Hardware Software Interface (External Interface Requirement in the SRS document) are analyzed as interface during the SRS requirements-based verification to achieve the hardware / software integration verification objective.

The test cases activity has the following inputs:

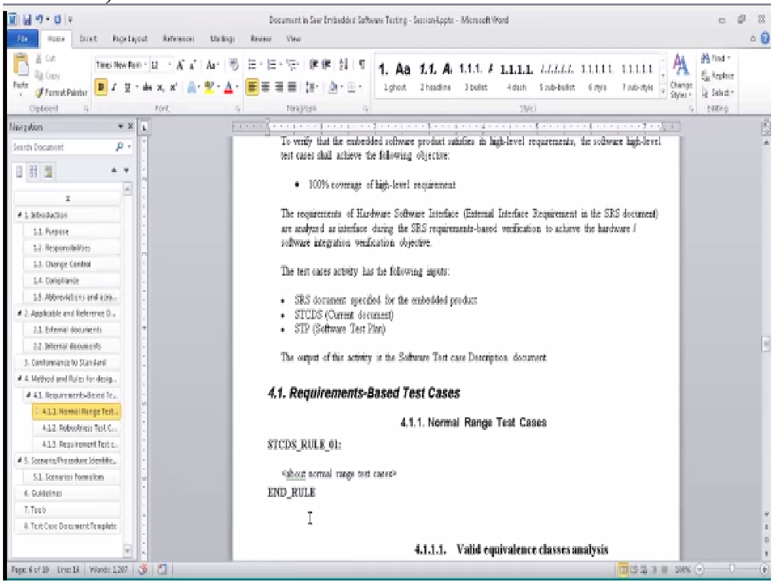
- SRS document specified for the embedded product
- STCDS (Current document)
- STP (Software Test Plan)

The output of this activity is the Software Test case Description document

4.1. Requirements-Based Test Cases

So all this strategies are embedded with the software test case, output of the test case is the software test case description document, that is test case document is embedded and it is test case development. Now coming to normal range test cases,

(Refer Slide Time: 18:06)



To verify that the embedded software product satisfies its high-level requirements, the software high-level test cases shall achieve the following objective:

- 100% coverage of high-level requirement

The requirements of Hardware Software Interface (External Interface Requirement in the SRS document) are analyzed as interface during the SRS requirements-based verification to achieve the hardware / software integration verification objective.

The test cases activity has the following inputs:

- SRS document specified for the embedded product
- STCDS (Current document)
- STP (Software Test Plan)

The output of this activity is the Software Test case Description document

4.1. Requirements-Based Test Cases

4.1.1. Normal Range Test Cases

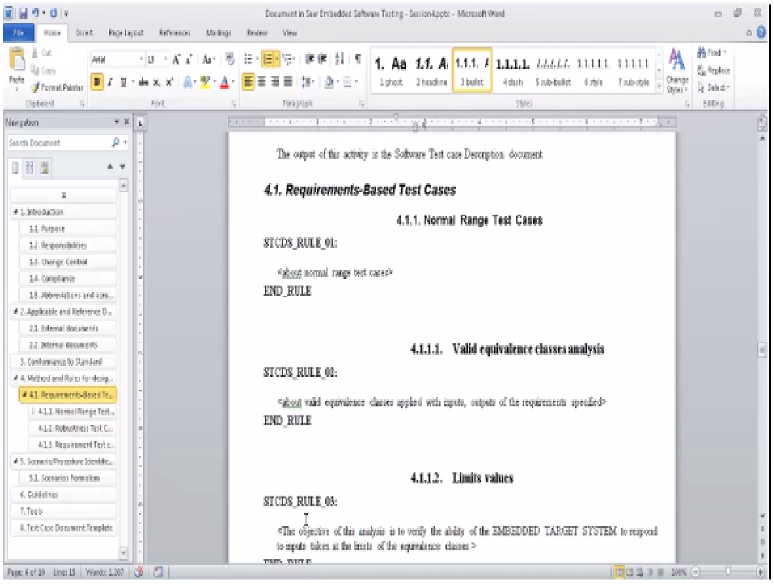
```
STCDS_RULE_01:

    <about normal range test cases>
END_RULE
```

4.1.1.1. Valid equivalence classes analysis

Basically the instruction identifies what are the methods that are used for requirement based test cases, there is normal range that is valid equivalence class that is the table based logic equation in special cases all this will be highlighted here this again depends on the target that is being under test.

(Refer Slide Time: 18:34)



The output of this activity is the Software Test case Description document

4.1. Requirements-Based Test Cases

4.1.1. Normal Range Test Cases

STCDS_RULE_01:

<about normal range test cases>

END_RULE

4.1.1.1. Valid equivalence classes analysis

STCDS_RULE_02:

<about valid equivalence classes applied with inputs, outputs of the requirements specified>

END_RULE

4.1.1.2. Limits values

STCDS_RULE_03:

<The objective of this analysis is to verify the ability of the EMBEDDED TARGET SYSTEM to respond to inputs taken at the limits of the equivalence classes >

We will go through some examples, normal range test cases will have normal range inputs of the normal range of testing for the testing requirements with that particular requirements allow the user to feed the values something like 1 to 10, the normal range is between one to 10 anything between 1 to 10 is called the normal range. This 1 to 10 is normal range, next as I said there should be a valid equivalence class just be analysed for that specific requirement.

So equivalence class will be applied on the inputs and also there is one point that is very important that the requirement can take one only input and based on the condition it could drive several output or multiple output, so how are you going to do it? How are you going to do the files? So we need to have consideration of the output also based on the input and other conditions it is very important.

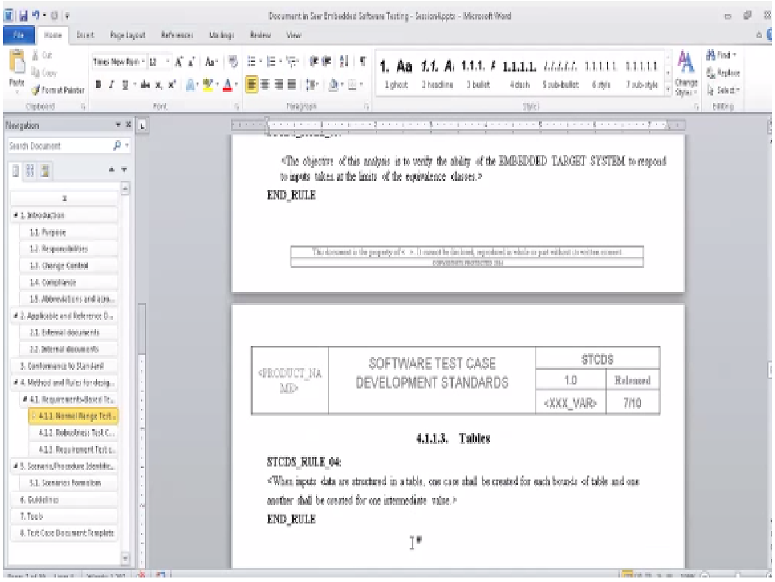
So the equivalence class should be applied on over the input, as I said yesterday the input could be 1, the input could be 5, somewhere 8, 10 etc. generally on the boundary cases sort of particular requirement is input or output, so one definitely has to differ 10 has to be left because these are coming on the quandral thus can be an intermediate family test type, similarly that output of the requirement basis saying 15 and 20.

So both needs to be test case, that means we should write test case impressive as that all the outputs that derived out of the requirements is fall into one of this test case design, the outputs suppose it is 15, 20, 25 three outputs are there we should mention that 15 and 25 are exercised definitely as a normal range and the midrange as the 20, so that we have exceled all this abnormal equivalence class.

Next we have a limit values that objective of this analysis to verify the ability of the embedded target system to respond to inputs taken at the limits of the equivalence class. What is that if there is an input of say 1, and the requirements is 1+ or – or point1 that means 10 % of that 1 is used then that needs to be exceled, that means we should have an ability to analyse the requirement in terms of the limit values that means the 1+ or – point1 in it could be 1- point1 into something, how much? -01 – 0.1, is 9 that it will take 1.1 so these needs to be used there are,

This is useful especially for the analog inputs where we use analog system which have a call transfers etc… it is very important aspect for limit values. So next we have tables suppose there are inputs mentioned with help of a table unexpected.

(Refer Slide Time: 23:16)



<The objective of this analysis is to verify the ability of the EMBEDDED TARGET SYSTEM to respond to inputs taken at the limits of the equivalence classes.>

END_RULE

This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

COPYRIGHT PROTECTED 2011

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	7/10

4.1.1.3. Tables

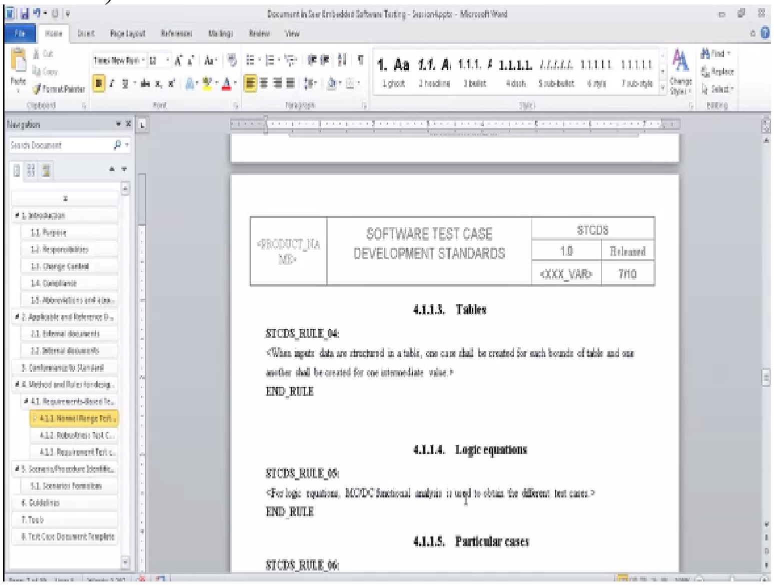
STCDS_RULE_04:

<When inputs data are structured in a table, one case shall be created for each bounds of table and one another shall be created for one intermediate value.>

END_RULE

Then at least one case should be created for the bounce of the table and one another repeated for an intermediate table, it is normal file saying that if the requirement is quanting the table, the table needs to be exceled in terms of either input or output based on each bound, these are lower bound upper bound and they should be an intermediate table aspect.

(Refer Slide Time: 23:50)



<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	7/10

4.1.1.3. Tables

STCDS_RULE_04:

<When inputs data are structured in a table, one case shall be created for each bounds of table and one another shall be created for one intermediate value.>

END_RULE

4.1.1.4. Logic equations

STCDS_RULE_05:

<For logic equations, MC/DC functional analysis is used to obtain the different test cases.>

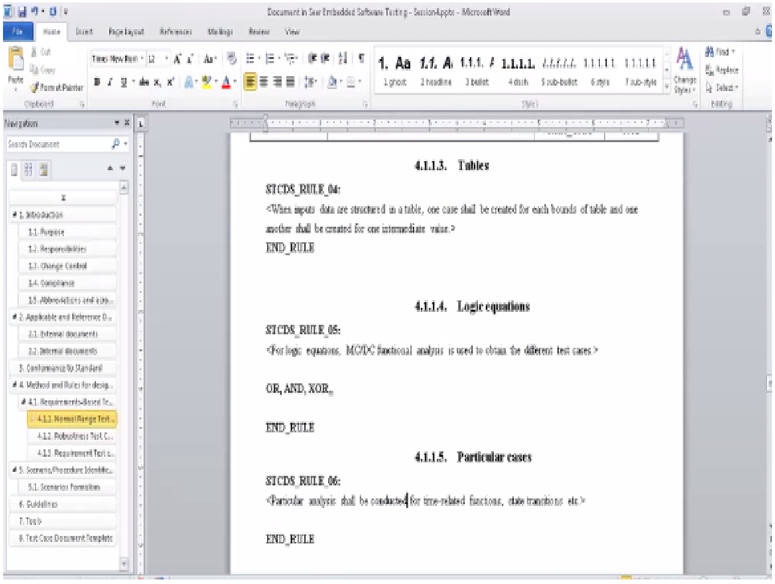
END_RULE

4.1.1.5. Particular cases

STCDS_RULE_06:

The next one is logic equations, there are very logical equations like usage of power and etc… we need to use MCDC function analysis which is used to obtain the different test case, I will tell you what is MCDC in later sessions this is one of the important aspects of embedded software testing, modified condition decision coverage I will explain what is that is to be considered specially for multiple inputs called multiple inputs OR, AND, XOR, then the special cases taken here on performance of the timing related

(Refer Slide Time: 24:50)


Document in Seer Embedded Software Testing - Session4.pptx - Microsoft Word
Navigation
1. Introduction
1.1. Purpose
1.2. Responsibilities
1.3. Change Control
1.4. Compliance
1.5. Abbreviations and acro...
2. Applicable and Reference D...
2.1. External documents
2.2. Internal documents
3. Conformance to Standard
4. Method and Rules for desig...
4.1. Requirements-Based Te...
4.1.1. Normal Range Test...
4.1.2. Robustness Test C...
4.1.3. Requirement Test c...
5. Scenario/Procedure Identific...
5.1. Scenarios Formalism
6. Guidelines
7. Tools
8. Test Case Document Template

4.1.1.3. Tables

STCDS_RULE_04:

<When inputs data are structured in a table, one case shall be created for each bounds of table and one another shall be created for one intermediate value.>

END_RULE

4.1.1.4. Logic equations

STCDS_RULE_05:

<For logic equations, MC/DC functional analysis is used to obtain the different test cases.>

OR, AND, XOR,,

END_RULE

4.1.1.5. Particular cases

STCDS_RULE_06:

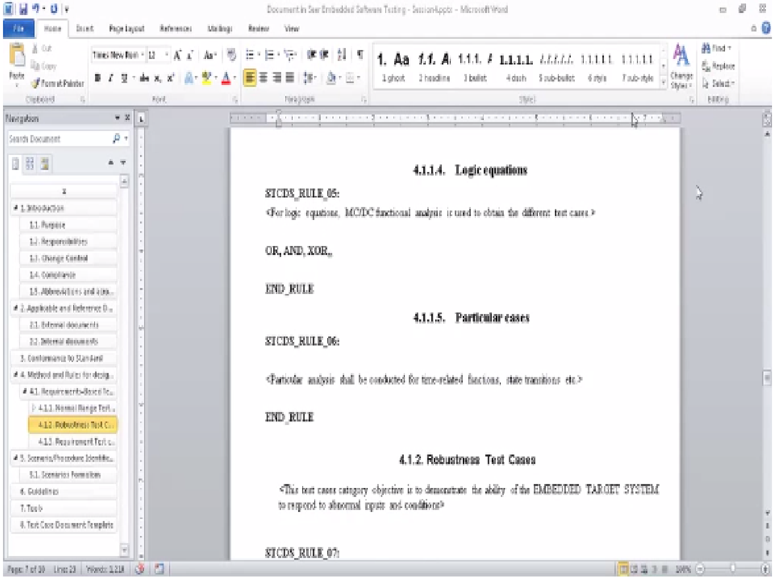
<Particular analysis shall be conducted for time-related functions, state transitions etc.>

END_RULE

For some requirement could say this product should remain and it is like time related functions there is the requirements corresponding to time related functions thus needs to be taken care on the performance basis or test basis development and that means state ambition onwards, I will try to put some minimum SRS for embedded software system will go through in times an SRS is to understand the requirement,

Probably all this can be derived SRS can have aspects we know how to state these for this having end moving to little states within the embedded software system. Suppose we have the input tested in which is able to see the robustness in terms of,

(Refer Slide Time: 26:04)



4.1.1.4. Logic equations

STCDS_RULE_05:

<For logic equations, MC/DC functional analysis is used to obtain the different test cases.>

OR, AND, XOR,,

END_RULE

4.1.1.5. Particular cases

STCDS_RULE_06:

<Particular analysis shall be conducted for time-related functions, state transitions etc.>

END_RULE

4.1.2. Robustness Test Cases

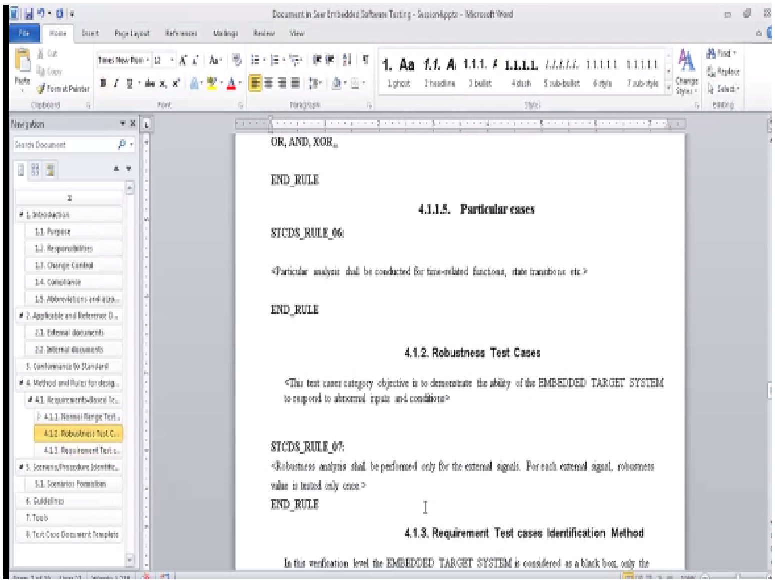
<This test cases category objective is to demonstrate the ability of the EMBEDDED TARGET SYSTEM to respond to abnormal inputs and conditions>

STCDS_RULE_07:

The 1+ or- 0.1+0.9 which is1.1 which should be system or it be requirement be embedded target it allows to provide robustness then we should be able to provide 0.8 or -1.0 probably many of this things may not allow that means we need to validate valuable I will explain that the instruction that definitely you should consider in terms of these in particular requirement it is understood that 1+ nor – 1.

Next we are going the robustness analysis,

(Refer Slide Time: 27:19)



OR, AND, XOR

END_RULE

4.1.1.5. Particular cases

STCDS_RULE_06:

<Particular analysis shall be conducted for time-related functions, state transitions etc.>

END_RULE

4.1.2. Robustness Test Cases

<This test cases category objective is to demonstrate the ability of the EMBEDDED TARGET SYSTEM to respond to abnormal inputs and conditions>

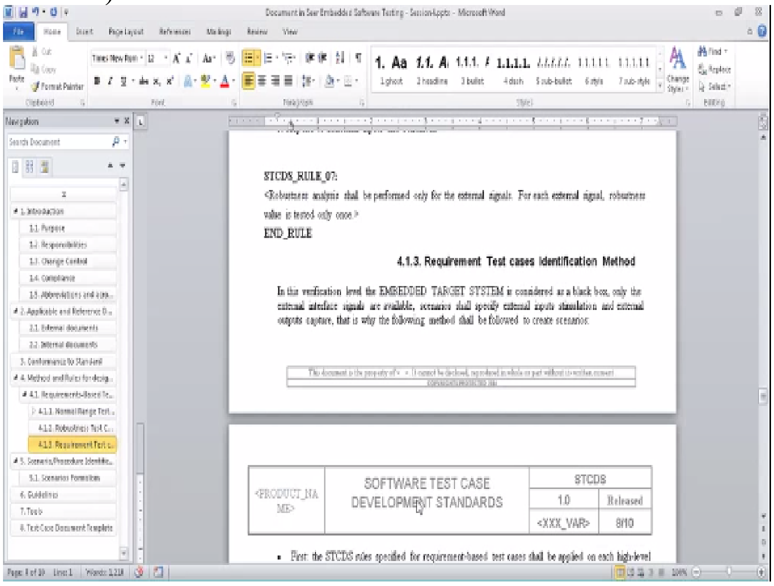
STCDS_RULE_07:
<Robustness analysis shall be performed only for the external signals. For each external signal, robustness value is tested only once.>
END_RULE

4.1.3. Requirement Test cases Identification Method

In this verification level the EMBEDDED TARGET SYSTEM is considered as a black box, only the

That be performed only for the external signals, for each external signal robustness value is tested only for code as I said this inputs for this particular requirement can be in terms of signals could be in different always as per the robustness

(Refer Slide Time: 27:43)



STCDS_RULE_07:
<Robustness analysis shall be performed only for the external signals. For each external signal, robustness value is tested only once.>
END_RULE

4.1.3. Requirement Test cases Identification Method

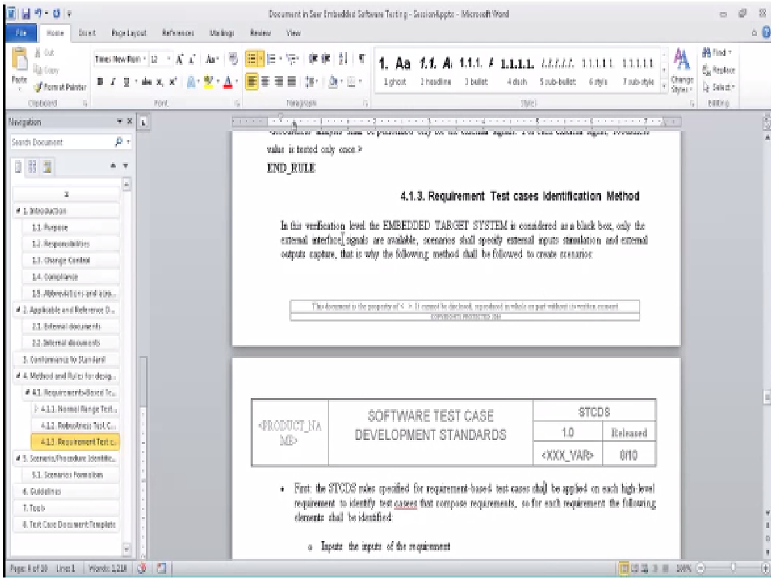
In this verification level the EMBEDDED TARGET SYSTEM is considered as a black box, only the external interface signals are available, scenarios shall specify external inputs stimulation and external outputs capture, that is why the following method shall be followed to create scenarios:

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	8/10

- First: the STCDS rules specified for requirement-based test cases shall be applied on each high-level

Then we have the requirement test cases identification method, in this verification level the embedded target system is considered as a black box only that is the high level requirements will be considered as a black box,

(Refer Slide Time: 28:02)


Document in Sear Embedded Software Testing - SessionAppts - Microsoft Word
File
Home
Insert
Page Layout
References
Mailings
Review
View
Navigation
Search Document
1. Introduction
1.1. Purpose
1.2. Responsibilities
1.3. Change Control
1.4. Compliance
1.5. Abbreviations and acro...
2. Applicable and Reference D...
2.1. External documents
2.2. Internal documents
3. Conformance to Standard
4. Method and Rules for design...
4.1. Requirements-Based Te...
4.1.1. Normal Range Test...
4.1.2. Robustness Test C...
4.1.3. Requirement Test c...
5. Scenario/Procedure Identific...
5.1. Scenarios Formalism
6. Guidelines
7. Tools
8. Test Case Document Template

value is tested only once.>

END_RULE

4.1.3. Requirement Test cases Identification Method

In this verification level the EMBEDDED TARGET SYSTEM is considered as a black box, only the external interface signals are available, scenarios shall specify external inputs stimulation and external outputs capture, that is why the following method shall be followed to create scenarios:

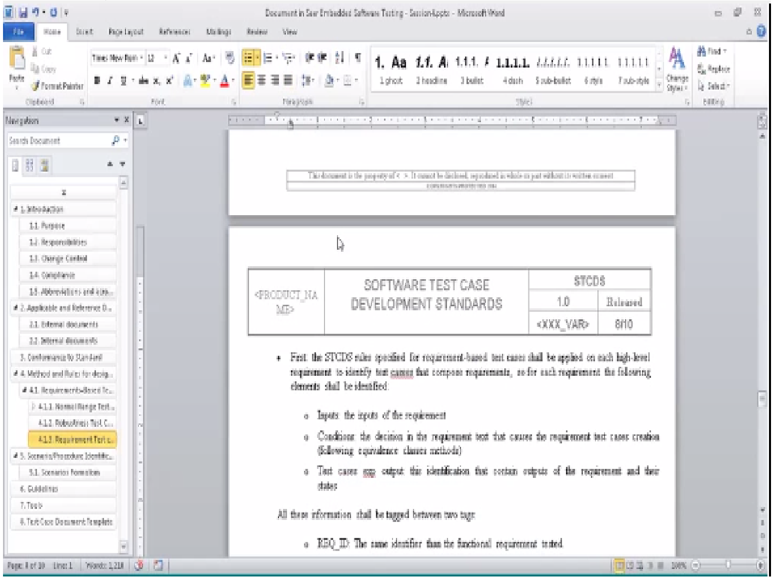
This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	8/10

- First the STCDS rules specified for requirement-based test cases shall be applied on each high-level requirement to identify test cases that compose requirements, so for each requirement the following elements shall be identified:
    - Inputs: the inputs of the requirement

It will have the scenarios shall specify essential aspects stimulation and external outputs that is why following method shall be followed to create sessions.

(Refer Slide Time: 28:33)


Document in Sear Embedded Software Testing - SessionAppts - Microsoft Word
Navigation
Search Document
1. Introduction
1.1. Purpose
1.2. Responsibilities
1.3. Change Control
1.4. Compliance
1.5. Abbreviations and acro...
2. Applicable and Reference D...
2.1. External documents
2.2. Internal documents
3. Conformance to Standard
4. Method and Rules for design...
4.1. Requirements-Based Te...
4.1.1. Normal Range Test...
4.1.2. Robustness Test C...
4.1.3. Requirement Test c...
5. Scenario/Procedure Identific...
5.1. Scenarios Formalism
6. Guidelines
7. Tools
8. Test Case Document Template

This document is the property of < >. It cannot be disclosed, reproduced in whole or part without its written consent.

<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	8/10

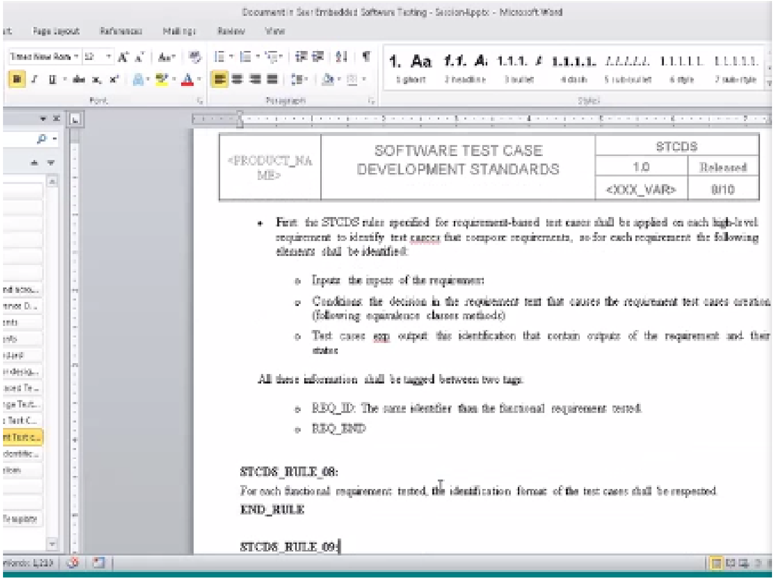
- First the STCDS rules specified for requirement-based test cases shall be applied on each high-level requirement to identify test cases that compose requirements, so for each requirement the following elements shall be identified:
    - Inputs: the inputs of the requirement
    - Conditions: the decision in the requirement test that causes the requirement test cases creation (following equivalence classes methods)
    - Test cases app output: this identification that contain outputs of the requirement and their states

All these information shall be tagged between two tags:

- REQ_ID: The same identifier than the functional requirement tested

The next one is first test case of the development standards the requirement based test cases shall be applied on each high level requirement that is every test case that fall into this standards, so the each requirement the following elements shall be identified, inputs the inputs of the requirement, condition the decision in the requirement test that causes the requirement test cases and equivalent slash P, I think in this format we have test case expected are based on this what are the output it can be identified all this information having there should be beginner and there should be ender in the test case.

(Refer Slide Time 29:33)



<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	0/10

- First the STCDS rules specified for requirement-based test cases shall be applied on each high-level requirement to identify test cases that compose requirements, so for each requirement the following elements shall be identified:
  - Inputs: the inputs of the requirement
  - Conditions: the decision in the requirement text that causes the requirement test cases creation (following equivalence classes methods)
  - Test cases exp output: the identification that contain outputs of the requirement and their status

All these information shall be tagged between two tags:

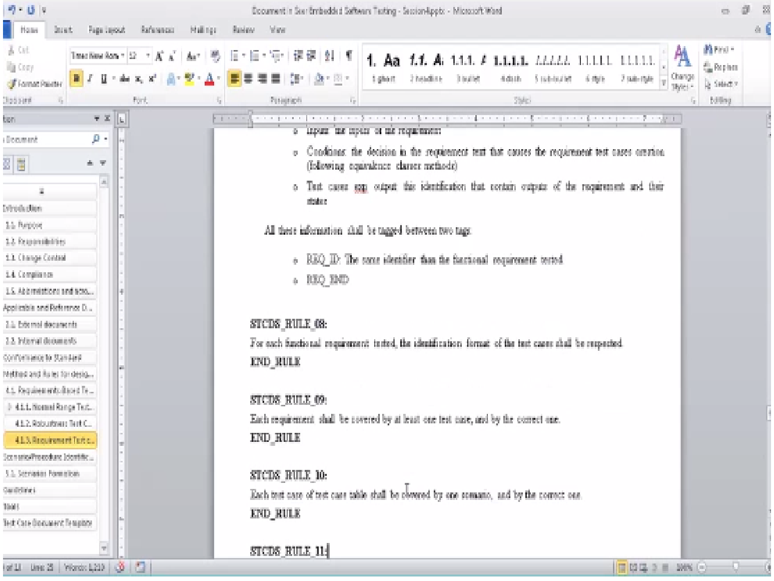
- REQ_ID: The same identifier than the functional requirement tested
- REQ_END

STCDS_RULE_08:
For each functional requirement tested, the identification format of the test cases shall be respected
END_RULE

STCDS_RULE_09:

The next one is, for each functional requirement tested by the equivalent format test case will be tested it should follow the {} this requirement shall be coverable at this one test case in the correct block. {} Every requirement needs to be test case what the way you wanted it is the automated way of {} for doing that there is an interface test case. Test case means they are actually they are one test case for each of the requirement one in the correct value.

(Refer Slide Time 30:15)



- Inputs: the inputs of the requirement
- Conditions: the decision in the requirement text that causes the requirement test cases creation (following equivalence classes methods)
- Test cases exp output: the identification that contain outputs of the requirement and their status

All these information shall be tagged between two tags:

- REQ_ID: The same identifier than the functional requirement tested
- REQ_END

STCDS_RULE_08:
For each functional requirement tested, the identification format of the test cases shall be respected
END_RULE

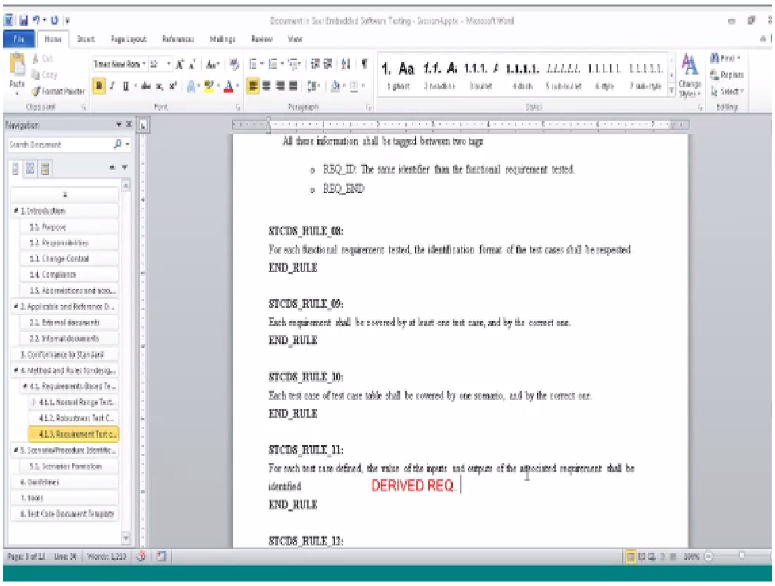
STCDS_RULE_09:
Each requirement shall be covered by at least one test case, and by the correct one.
END_RULE

STCDS_RULE_10:
Each test case of test case table shall be covered by one scenario, and by the correct one.
END_RULE

STCDS_RULE_11:

The next one is each test case covered by scenario and by the correct value, it turn over whatever will come under that, now the test value that are identify for each test case will be find, the value of the input of the associated requirement identify, there any supported requirement that has to be identify, typical example will be the read the value of derived requirement, the derived requirement something like sometime what will happen? We are talking about that derived requirement or supporting requirement, what will happen means the primary requirement are sufficient, so address various from the functionality it need to have a supporting requirement of, it is the level.

(Refer Slide Time 30:55)



All these information shall be tagged between two tags

- REQ_ID: The same identifier than the functional requirement tested.
- REQ_END

STCDS_RULE_08:

For each functional requirement tested, the identification format of the test cases shall be respected

END_RULE

STCDS_RULE_09:

Each requirement shall be covered by at least one test case, and by the correct one.

END_RULE

STCDS_RULE_10:

Each test case of test case table shall be covered by one scenario, and by the correct one.

END_RULE

STCDS_RULE_11:

For each test case defined, the value of the inputs and outputs of the associated requirement shall be identified DERIVED REQ.

END_RULE

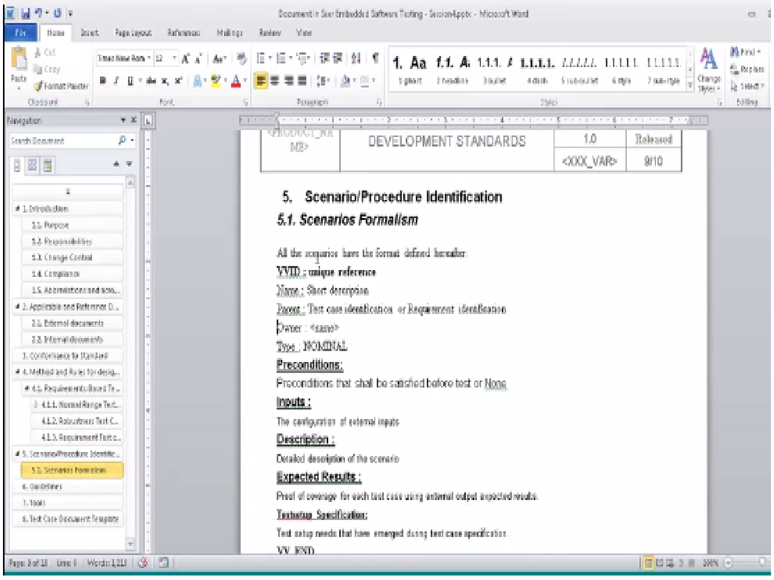
STCDS_RULE_12:

Derived requirement, something like we have a multiple choice instructions like detail, what will happen, when primary requirement it defect the complaint, what will do is, we will highlight in the primary requirement. These are the instruction that have been used. In the supporting and derived requirement, we are mention that this instruction will return the value of this right,

Why? This derived requirement is are implemented, because consolation of the requirement is very important, this requirement have to be unique, it cannot be handled that means, it cannot have multiple requirement, that is a brilliant role, but it has to be initialized, actually it has the one identification requirement, it does not have the multiple requirement, then it has to identify the next requirement, that is what supporting requirement for this reason with a derived data.

Basically each of the test case should be qualified as more, primarily this mean type of a testing is progress, this something that I have discuss we will know about the supporting requirement and derived requirement, so some test case cannot executed, which would have a enough space, identifying that has in the test cases, we will define how it will be preview?

(Refer Slide Time 33:43)



<PRODUCT_NAME>	DEVELOPMENT STANDARDS	1.0	Released
		<XXX_VAR>	9/10

5. Scenario/Procedure Identification

5.1. Scenarios Formalism

All the scenarios have the format defined hereafter:

VVID : unique reference

Name : Short description

Parent : Test case identification or Requirement identification

Owner : <name>

Type : NOMINAL

Preconditions:

Preconditions that shall be satisfied before test or None

Inputs :

The configuration of external inputs

Description :

Detailed description of the scenario

Expected Results :

Proof of coverage for each test case using external output expected results.

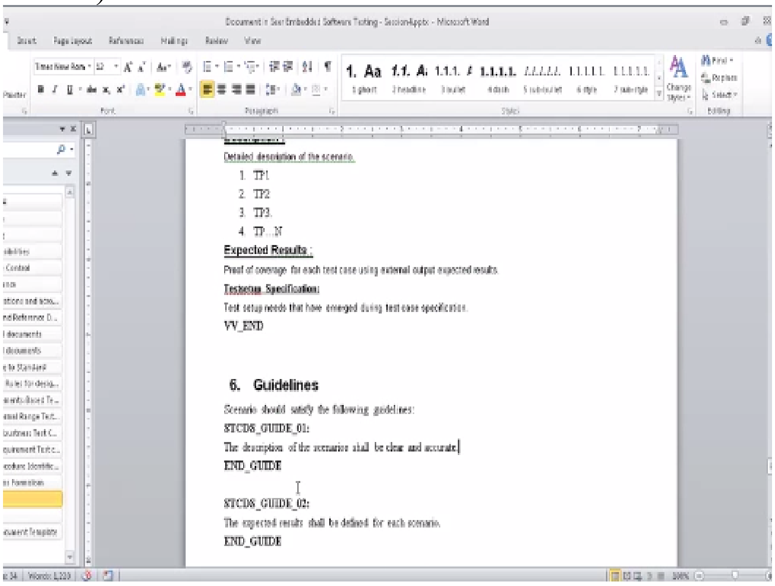
Testsetup Specification:

Test setup needs that have emerged during test case specification

VV_END

The next one is a closing production, if it is done through an example test over here, we will have to identify the name by which specification is done, what is the preview condition and the uses, which is to identify the inputs, it has to test the requirement, this should be where description or it can add all those steps that is required of, in terms of strategy planning , we can use the excel sheet, because it has to specify the tools, it has to suggested to have the tools and especially it has to modify the accessing unit, that is what the user is determine to the system, it has to highlight each test case to the related system and procedures has the proper specification it is consider by the system procedure, guidelines has the certain rules to follow the instructions, guide lines are effecting from requirement for their need, they will be consider as one by one, guide lines are depreciative, based on something like initialized to guide lines

(Refer Slide Time 36:05)



Detailed description of the scenario.

1. TP1
2. TP2
3. TP3.
4. TP…N

Expected Results :

Proof of coverage for each test case using external output expected results.

Testexm Specification:

Test setup needs that have emerged during test case specification.

VV_END

6. Guidelines

Scenario should satisfy the following guidelines:

STCDS_GUIDE_01:

The description of the scenario shall be clear and accurate

END_GUIDE

STCDS_GUIDE_02:

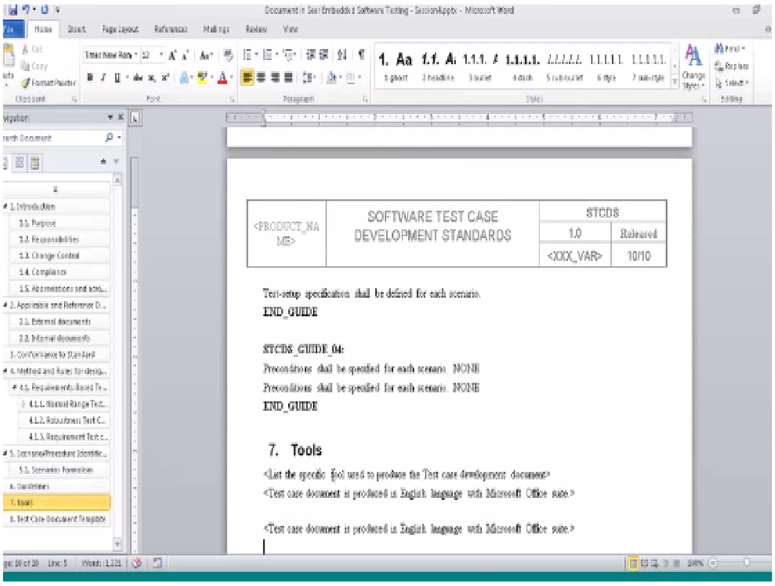
The expected results shall be defined for each scenario.

END_GUIDE

For example, the description of scenario should be clear, the accepted result should be for each scenario, scenario should be result in a respective form, the set of specification is identify by the scenario, these scenario has different aspects, finally the scenario will be executed with this conditions. Some of the requirements are do not have specific condition, directly can execute. It is based on actual condition, in case it has no preview condition, it has to develop the test case development, you can do a document like alignment or exercise that are used in this system, in this system testing tool can be highlight, test case has a template document,

For an example, test case of the description is the same, test case has different requirement, the document will destroy all the test case during in the process, manually the test case is a part of the coverage, the test case have to interact with reference documents, we have gone through test case example, test plan, test spec, test case design, and test standards, with help of that this test will be design, we have a simple example for test case, we all know that white board requirement right, white board requirement it is the sample test of this system,

(Refer Slide Time 36:40)



<PRODUCT_NAME>	SOFTWARE TEST CASE DEVELOPMENT STANDARDS	STCDS	
		1.0	Released
		<XXX_VAR>	10/10

Test-setup specification shall be defined for each scenario.
END_GUIDE

STCDS_GUIDE_04:
Preconditions shall be specified for each scenario. NONE
Preconditions shall be specified for each scenario. NONE
END_GUIDE

7. Tools

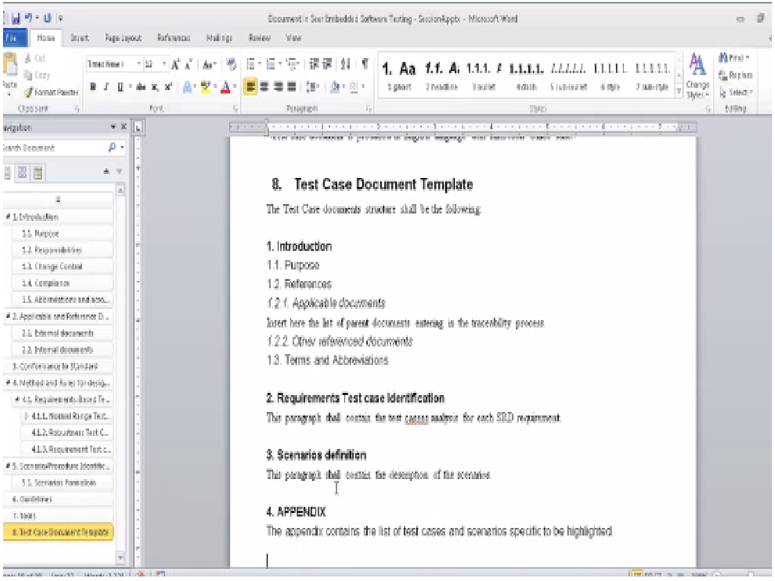
<List the specific tool used to produce the Test case development document>
<Test case document is produced in English language with Microsoft Office suite.>

<Test case document is produced in English language with Microsoft Office suite.>

So requirements are something like white board, shall we had a length of the feedback, so the white board will allow to write the colour flame, that is why it is called white board, so what are the test case? Test case should be identify, I will say, measure the white board and verify that use to measure width of the 3 feet and 4 feet so this will cover test requirements. Next test case could be one can able to write on the board that is a next test case because we should allow first to write so that is test case.

The next requirement we have to tested with check whether one can able to write black colour in this board, we can use these file in software so that could be better word verifying that already we have a verified statement, either from the white board, the next test case is on check whether we written that in black and green are physic, that mean there is a difference you should not write that is the first. Next thing will be visible both are very important and also I think you can add to the something like that, verify whether thumbs and sizes are what is supplement to this test you have specified here black.

(Refer Slide Time 37:13)



8. Test Case Document Template

The Test Case documents structure shall be the following:

1. Introduction
1.1. Purpose
1.2. References
1.2.1. Applicable documents
Insert here the list of parent documents entering in the traceability process
1.2.2. Other referenced documents
1.3. Terms and Abbreviations

2. Requirements Test case Identification
This paragraph shall contain the test cases analysis for each SRD requirement.

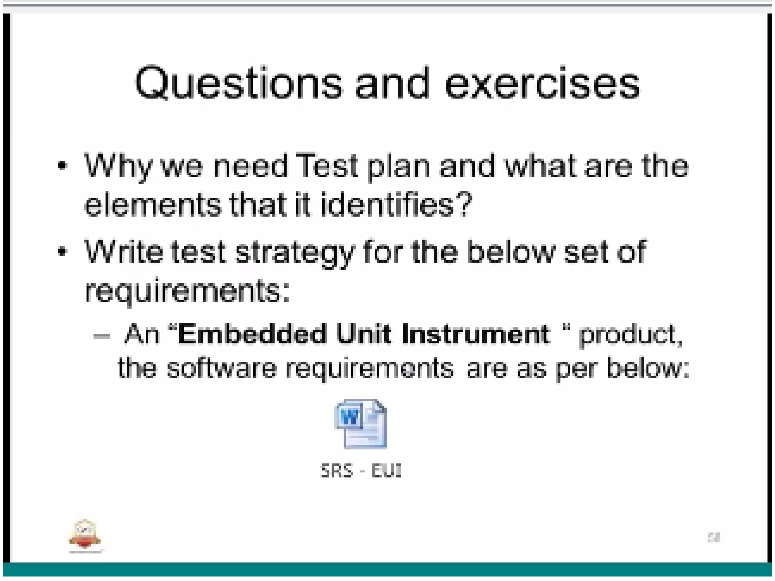
3. Scenarios definition
This paragraph shall contain the description of the scenarios.

4. APPENDIX
The appendix contains the list of test cases and scenarios specific to be highlighted.

So you can specify this colour similarly we can rewrite green so and of course as I said you can have the combination of both. These both should be allowed right, which is that, here I taking a simple example so you can analysis this in terms of a software example or else from the format from the embedded software system. Verify whether one from right B and C code and that could be checked whether or verified whether one can visualize the written blue and green letters, word, picture actually this is not a good practice to have or from the letter or word process better you can mention as analysis test case relating just word.

For this there is an example, next test case is check whether we return in the black board try to eraser the words in the interesting aspectives because requirement as not told but this is the intention where you can have a generalized requirement because we know that white board we are testing this so this test case is interesting because of that. Try to erase the words written and write in the white board; verify that the words are erased I think we have written erase is there. Note that we have already done in terms what will happen is the intuition user may add more requirements because we have a good knowledge on the system if a system is white board. So during the review what will happen is this will result in the modifying the requirement assume that or this will result in adding further requirement so in that way as I mentioned yesterday test key or the tester should be independent. So it can think out of the bar we can think from this perspective so this is always understood. So this is example test case any doubts are anything is there in this you can ask me.

(Refer Side Time 43:47)



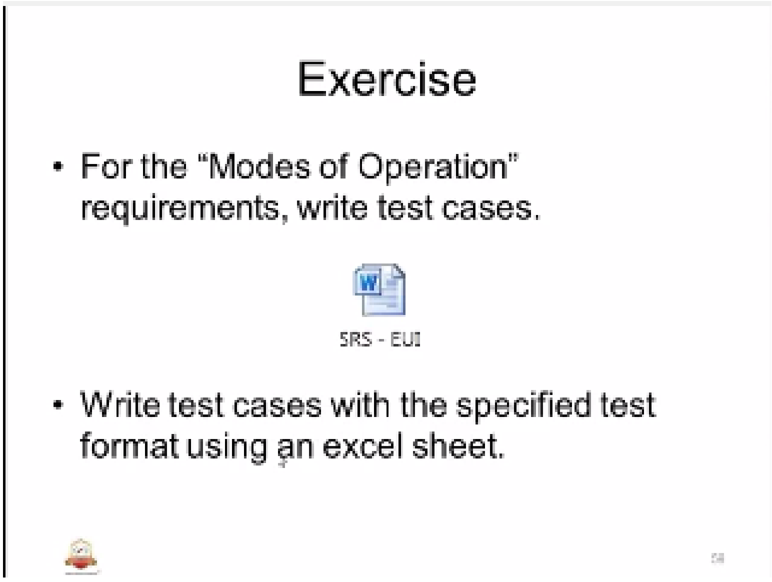
Questions and exercises

- Why we need Test plan and what are the elements that it identifies?
- Write test strategy for the below set of requirements:
  - An "**Embedded Unit Instrument** " product, the software requirements are as per below:

SRS - EUI

Let's come to, having gone through all this test case standards, test case specification, plan I have taken position simple query in the site may be you can reduce and edit sentences on since we have got are not. Why we need test plan and what are the element that it identifies? It is a simple question write test strategy for the below set of requirement. You know what is test strategy? By now and we should write test strategy for testing the below requirement, below requirements is attached here. An embedded unit instrument just put an example "EUI" product the software requirements are as per below. So you should have a test strategy for the test plan, I will further. What is the next question?

(Refer Side Time 44:39)



Exercise

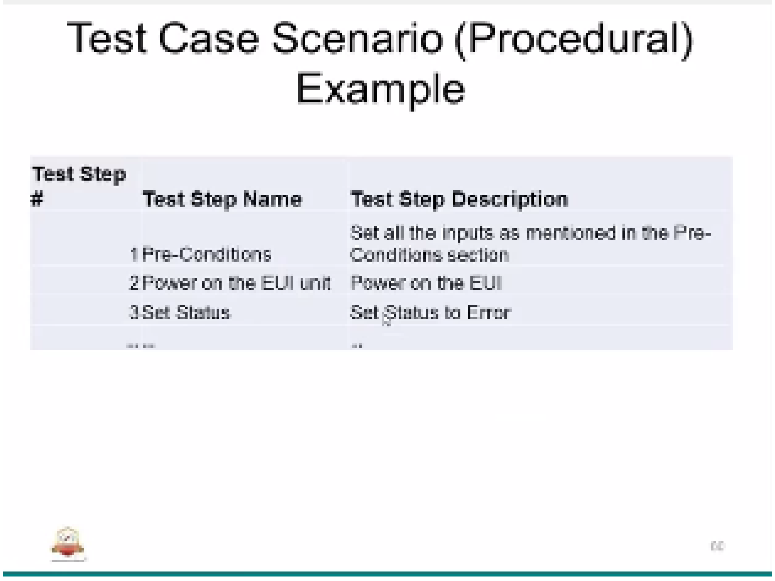
- For the "Modes of Operation" requirements, write test cases.

SRS - EUI

- Write test cases with the specified test format using an excel sheet.

And together we can see the entire, the examples and the SRS has modes of operation one set of the format. So next session is arrive the test cases and modes of operation, there are different operation requirement that is for now you can write test cases for that modes of operation and it will see how good we are and we can write test cases with the specified test format which should be read and left side it is a specific device.

(Refer Side Time 45:12)

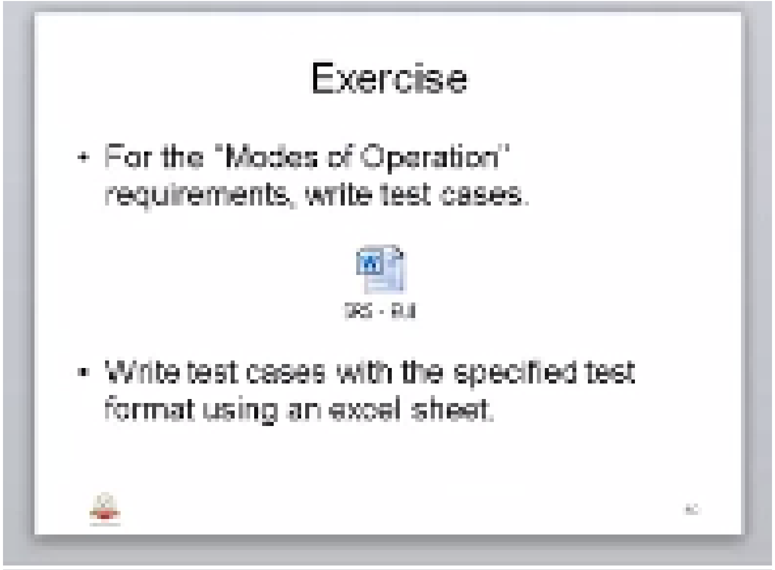


Test Case Scenario (Procedural) Example

Test Step #	Test Step Name	Test Step Description
1	Pre-Conditions	Set all the inputs as mentioned in the Pre-Conditions section
2	Power on the EUI unit	Power on the EUI
3	Set Status	Set Status to Error
	... ...	..

Excel sheet can be complained this is of course an example of a test case scenario or procedural example this is the test step one, two, three etc.. test step name this highlight what I mean in the description I want to tell that, set all the inputs as mentioned in the per condition section. So pre condition means it will be pointing to test section which will specify the condition that are required for executing this particular test. So after that we are going to power on the unit then we set test status or error it could be depending on the set status to error because you want to test something that are from the notes of unconditional where excel to go we can write using that condition I will share which is the examples based on that you can understand.

(Refer Side Time 46:21)



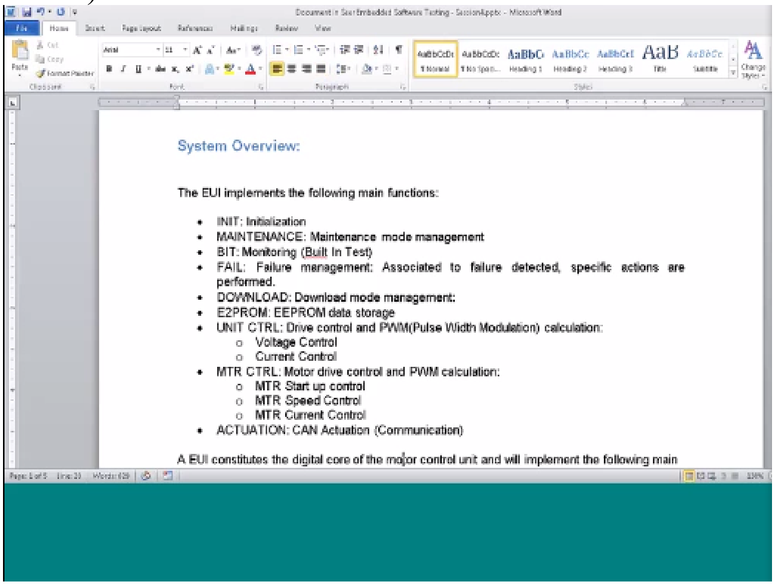
Exercise

- For the "Modes of Operation" requirements, write test cases.

- Write test cases with the specified test format using an excel sheet.

Question basically we are doing in the format write test strategy first we need to write the test strategy how I am going to address the complete requirement we need to understand the complete requirement. Once you understood there are SRS there is one section called modes of operation.

(Refer Side Time 47:02)



System Overview:

The EUI implements the following main functions:

- INIT: Initialization
- MAINTENANCE: Maintenance mode management
- BIT: Monitoring (Built In Test)
- FAIL: Failure management: Associated to failure detected, specific actions are performed.
- DOWNLOAD: Download mode management:
- E2PROM: EEPROM data storage
- UNIT CTRL: Drive control and PWM(Pulse Width Modulation) calculation:
  - Voltage Control
  - Current Control
- MTR CTRL: Motor drive control and PWM calculation:
  - MTR Start up control
  - MTR Speed Control
  - MTR Current Control
- ACTUATION: CAN Actuation (Communication)

A EUI constitutes the digital core of the motor control unit and will implement the following main

We need to write test cases and you can use the template so this is excel you can open the SRS, so you can go through this you will have a requirements. So basically SRS will have system overview, I had to put myself a simplest requirement it is called the embedded unit instrument. So overview of this embedded unit instrument which as the below functionality initialization maintenance and maintenance mode management they we have a monitoring; monitoring built in test, you have to solve that built in test in the unit instrument other serial management with referring mode.

Which will have all the failures detected the actions that are recorded as a result of reduce, that also part of the failure nod. Then we have download mode management I told you that embedded software needs to be downloaded on to the target. So the software should have a program I mean the software should have a condition which will accept this downloadable program, usually the embedded system target will have two sets of first system one is root loader other one is an application. Usually there will be changes in the application. Root software + loader = root loader, this is responsible for downloading the new program. New program could be, new program or image= application from this system we will have programmable root, root loader aspect. Because we want to upload or download new loader itself so what we will do we will have to root software.

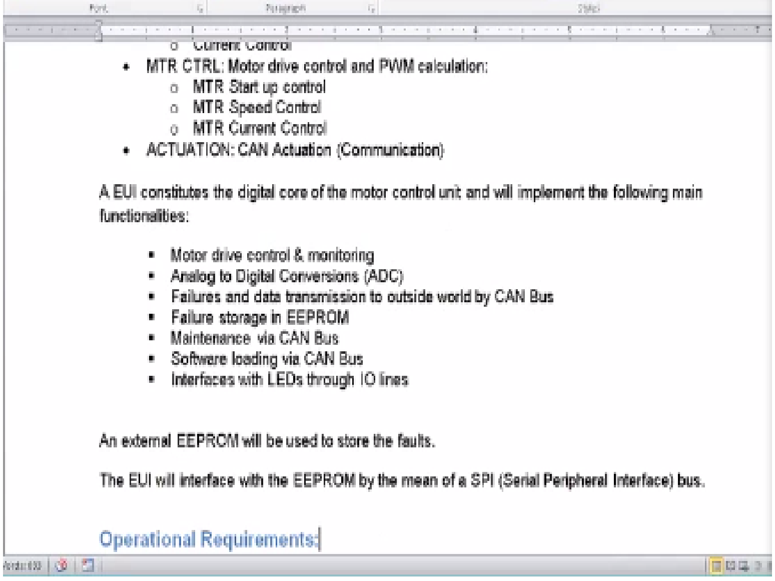
So that is having downloaded in load management which is responsible for downloading the in loader. So basically download management it is a SRS function which are part new program and SRS function part of that also used aspects. Next one is register form so register form I will not go into details of the register for mention. Part which is seeing in the embedded system it is programmable memory we do not get into it.

The external world which can be access to such many of the data or user wants to program the configuration of calibration. So this needs to find in between in the separate device, this is part of the embedded system. It is a register is a fault that needs to be stored, when the system is placed in the memory which identifies the EEPROM or the E2PROM that will have in the register form. Then user typical embedded application unit control, this embedded unit are the control function which will drive the which will drive and calculate the PWM pulse width modulation and details of the I will not go because which is pulse width modulation.

So this is the requirement or the functionality of the system how you can drive the target, while driving the way has a different functionality voltage control and current control. Then we have a specific motor control MTR and CPRM so we should have to drive and control PWM calculation so it is a motor start or motor speed or motor current so I tried to put the motor there are further complexities involved so you just highlight of the basic requirement of the functionality software physical embedded software.

There is an actuation requirement which is for communication, that will identify the inputs are the outputs how it is going to be communicated with the embedded software system. So that is with the help of scan,

(Refer Side Time 52:29)



- Current Control
- MTR CTRL: Motor drive control and PWM calculation:
  - MTR Start up control
  - MTR Speed Control
  - MTR Current Control
- ACTUATION: CAN Actuation (Communication)

A EUI constitutes the digital core of the motor control unit and will implement the following main functionalities:

- Motor drive control & monitoring
- Analog to Digital Conversions (ADC)
- Failures and data transmission to outside world by CAN Bus
- Failure storage in EEPROM
- Maintenance via CAN Bus
- Software loading via CAN Bus
- Interfaces with LEDs through IO lines

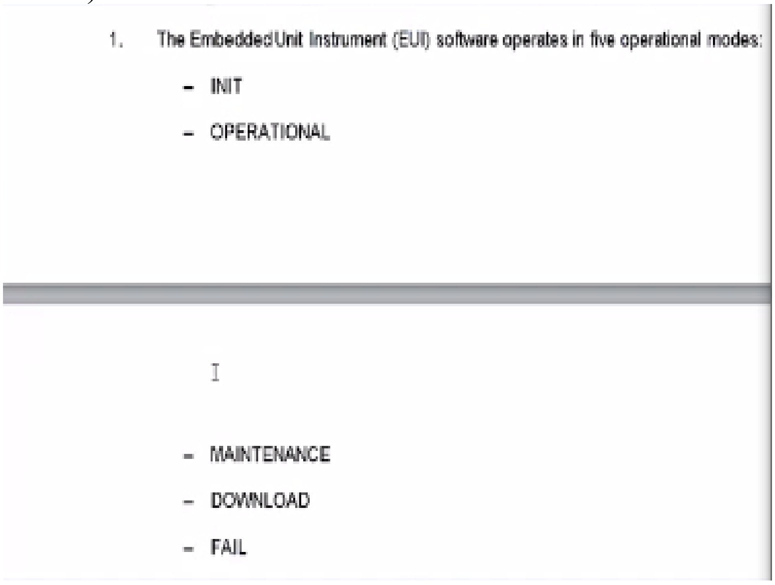
An external EEPROM will be used to store the faults.

The EUI will interface with the EEPROM by the mean of a SPI (Serial Peripheral Interface) bus.

Operational Requirements:

Something is highlighted to the embedded unit these are four of the motor control unit and motor drive control and monitoring, analog to digital conversions, failures and data transmission to outside world by CAN bus, failure storage in the memory, and many maintenance that required embedded system, software loading parallel management with the help of CAN competition, and there are of course two LEDs with the help of IO line that would be interface. The next external will be used to store fault, the EUI will interface with the EEPROM by the mean of a SPI bus, so that is what operation requirement interface integration so there are different thing involved in the operation requirement which should identify operation requirement it will identify current initialization to identify download etc.. Accordingly you can see the picture. Operation requirement,

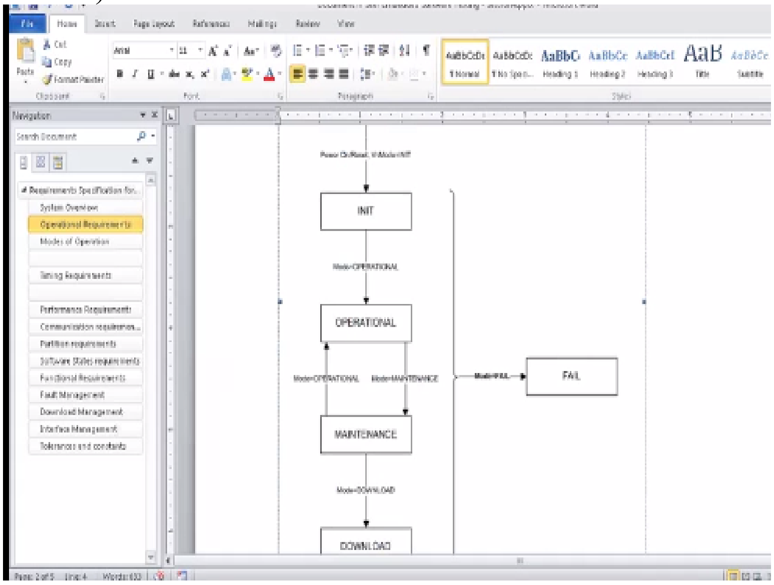
(Refer Side Time 54:04)



1. The Embedded Unit Instrument (EUI) software operates in five operational modes:
   - INIT
   - OPERATIONAL
   - MAINTENANCE
   - DOWNLOAD
   - FAIL

These are the operation requirement, with a small defection of the different nod these are the operation,

(Refer Slide Time 54:17)



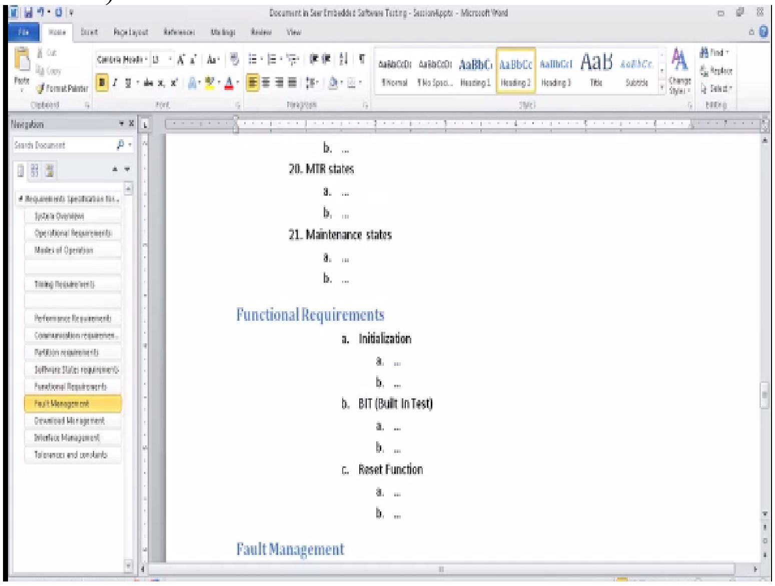
So there is a unit mode, there is an operation mode, there is a maintenance mode, download, and up on schedule it is a user mode. So these are some of the fore view of operational requirements, so this is the operational requirement with the operation this is the operation requirement of mode.

Coming to the modes of operation what are the modes? That embedded units have not gone through. So these are the requirements you can go through embedded there they can explain in this session.

So what are the requirements for different mode? Unit mode, maintenance mode, then we have download etc… all this cover in construct of the embedded target, so the timing requirements there are communication requirements, there are partition requirements there is no separate applications involved from the embedded target only one single application is done, some industries they call as well as application it is not integrated in to the application, so different states embedded in the instrument, so what are the steps? What is the most test numbers?

What is the most test numbers based on this software states requirements, then we have functionality.

(Refer Slide Time: 56:12)



b. ...

20. MTR states
    a. ...
    b. ...

21. Maintenance states
    a. ...
    b. ...

Functional Requirements

a. Initialization
    a. ...
    b. ...

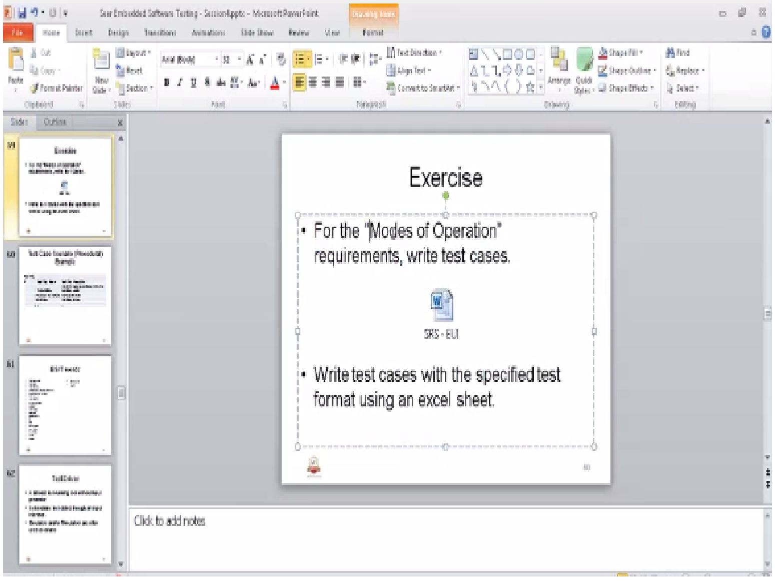
b. BIT (Built In Test)
    a. ...
    b. ...

c. Reset Function
    a. ...
    b. ...

Fault Management

Again how we are going to do, initialization is there, built in test is there, reset function is there then we have fault management download management, tolerance and constants, as I said some of the tolerances that needs to be identified, so what you need to do is?

(Refer Slide Time: 56:44)

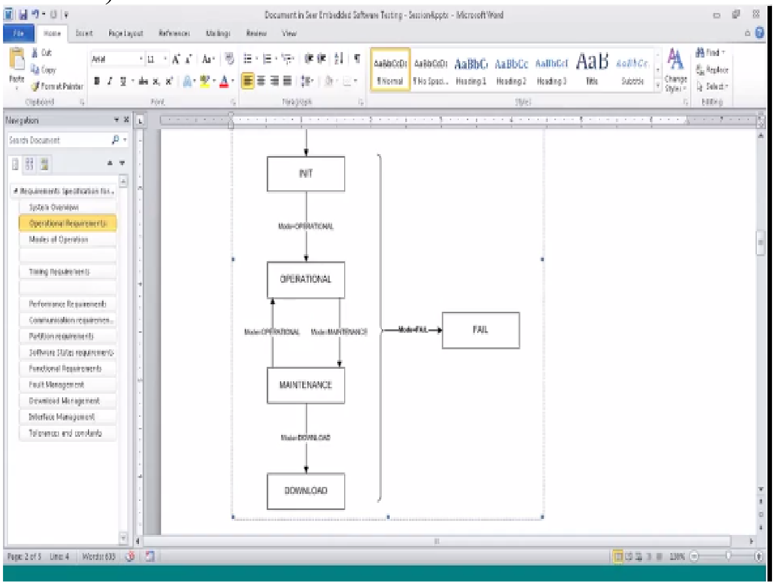

Saer Embedded Software Testing - SessionAppts - Microsoft PowerPoint
Click to add notes
SRS - EUI

Exercise

- For the "Modes of Operation" requirements, write test cases.
- Write test cases with the specified test format using an excel sheet.

For the modes of operation for we saw just now we need to write test case, that is here based on the modes of operation 1, 2, 3, ,4, 5 operations, we need the maintenance, download, fail, so the embedded unit instrument should be one of this should be one this is small lecture and more bale to be insulation then insulation to be fine they will be operational when the maintenance is required and use of the download is required for downloader mode while doing the operation ,

(Refer Slide Time: 57:42)



Now let us go to the next term comes in to the operational the mode should be operational within here, if the mode is simply like an aspect it is updated by the embedded software, there is a five requirements that is we do here test case design, test case scenario and test case standard so we have embedded unit instrument, test plan, we have test case steps, we have test case descriptions etc…