DEPT. OF MATHEMATICS & COMPUTING

LABORATORY LOCATION

(1) SAP- DRS LAB I (2) SAP- DRS LAB II (3) DST- FIST LAB

LAB MANUAL FOR SOFTWARE ENGINEERING

Course Code MCC401

Prof. S. Chatterjee

&

Prof. N Jana

IIT (ISM) Dhanbad

Do’s and Don’ts:

1. Come to the lab. in time. After 5 minutes of entry time nobody will be allowed.

2. Follow the Instruction provided by the Faculty/Research Scholar at the beginning.

2. Write the program using the instructions given and then start coding.

3. Don’t take out the computer accessories like: mouse, patch chord, etc.

4. During the Lab. hour don’t create any disturbance

Software Requirement:

1. C
2. C++
3. JAVA

Reference books:

1. Software Engineering: A Practitioner’s Approach by R.S. PressMan, McGraw Hill.

2. Fundamentals of Software Engineering by R. Mall, Prentice Hall of India.

3. An Integrated Approach to Software Engineering by P. Jalote, Narosa.

Lab Examination:

After completion of all the programs and project work in the given list of experiments there will be **one** lab exam of 100 marks during the slot provided in the academic calendar. Examination will be based on Programming followed by Viva Voce.

The 100 marks will distributed as follows:

60 Marks in program writing and execution, 30 marks in Lab. File, 10 marks in Viva Voce.

LIST OF PROGRAMS

1. Implementation of COCOMO Model for cost estimation
2. Implementation of COCOMO Model for time estimation
3. Implementation of Halstead metric.
4. Implementation of CPM.
5. Implementation of PERT.
6. Preparation of SRS for some defined project.
7. Development of structural design for some defined project.
8. Development of object model using UML techniques for some defined project.
9. Design of Test suits for Black Box Testing.
10. Design of Test Suits for White Box Testing.

EXPERIMENT-1

Aim: Implementation of COCOMO Model for cost estimation

Tools/Apparatus: None

Procedure:

1) Input the number of lines of code
2) Predict the type of project – organic, semi-detached or embedded
3) Calculate the effort applied, average staff size, productivity and cost

EXPERIMENT-2

Aim: Implementation of COCOMO Model for time estimation

Tools/Apparatus: None

Procedure:

1) Input the number of lines of code
2) Predict the type of project – organic, semi-detached or embedded
3) Calculate the effort applied, development time

EXPERIMENT-3

Aim: Implementation of Halstead metric

Tools/Apparatus: None

Procedure:

1) Write a short program in gedit mode
2) Call the program in the main Halstead program
3) Compute total number of operators and operands
4) Find the number of unique operators and operands
5) Calculate program vocabulary, program length, volume, difficulty and effort

EXPERIMENT-4

Aim: Implementation of CPM

Tools/Apparatus: None

Procedure:

1) Draw the activity network diagram
2) Calculate the earliest start time, earliest finish time, latest start time and latest finish time
3) Compute total float and free float
4) Find the critical activities and critical path
5) Calculate the project completion time

EXPERIMENT-5

Aim: Implementation of PERT

Tools/Apparatus: None

Procedure:

1) Draw the activity network diagram
2) Calculate expected time or average time using optimistic time, most likely time and pessimistic time
3) Compute variance and standard deviation
4) Calculate the earliest start time, earliest finish time, latest start time and latest finish time
5) Compute total float and free float
6) Find the critical activities and critical path
7) Calculate the probability of completion of project

EXPERIMENT-6/ Project 1

Aim: Preparation of SRS for projects like flight reservation/ ATM transaction

Tools/Apparatus: None

Procedure:

1) Create an Outline (Or Use an SRS Template). Your first step is to create an outline for your software requirements specification.
2) Start With a Purpose.
3) Give an Overview of What You'll Build.
4) Detail Your Specific Requirements.
5) Get Approval for the SRS.

EXPERIMENT-7 / Project 2

Aim: Development of structural design for projects like admission in schools or colleges/ hospital management

Tools/Apparatus: None

Procedure:

1) The structural view diagram: Class diagram, object diagram.

2) Identify various elements such as classes, member variables, member functions etc. of the class diagram.

3) Draw the class diagram as per the norms.

4) Identify various elements such as various objects of the object diagram.
5) Draw the object diagram as per the norms.

EXPERIMENT-8 / Project 3

Aim: Development of object model using UML techniques for projects like online shopping/ airport check-in and security screening

Tools/Apparatus: None

Procedure:

1) Identify various elements of the system to be drawn using the IDE.

2) Use the UML options of the MS Visio software to draw like Component diagram and deployment diagram.

EXPERIMENT-9a / Project 4

Aim: Design of Test suits for Black Box Testing

Tools/Apparatus: None

Procedure:

1) Initially, the requirements and specifications of the system are examined.

2) Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.

3) Tester determines expected outputs for all those inputs.

4) Software tester constructs test cases with the selected inputs.

5) The test cases are executed.

6) Software tester compares the actual outputs with the expected outputs.

7) Defects if any are fixed and re-tested.

Example:

Regression tests – Regression techniques is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

EXPERIMENT-9b / Project 5

Aim: Design of Test suits for Black Box Testing

Tools/Apparatus: None

Procedure:

1) Initially, the requirements and specifications of the system are examined.

2) Tester chooses valid inputs (positive test scenario) to check whether SUT processes them correctly. Also, some invalid inputs (negative test scenario) are chosen to verify that the SUT is able to detect them.

3) Tester determines expected outputs for all those inputs.

4) Software tester constructs test cases with the selected inputs.

5) The test cases are executed.

6) Software tester compares the actual outputs with the expected outputs.

7) Defects if any are fixed and re-tested.

Example:

Equivalence Class Testing: It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.

EXPERIMENT-10a / Project 6

Aim: Design of Test Suits for White Box Testing

Tools/Apparatus: None

Procedure:

1) Identify the feature, component, program to be tested.

2) Plot all possible paths in a flow graph.

3) Identify all possible paths from the flow graph.

4) Write Test Cases to cover every single path on the flow graph.

5) Execute, rinse, repeat.

Example:

White Box penetration techniques: In this testing, the tester/developer has full information of the application's source code, detailed network information, IP addresses involved and all server information the application runs on. The aim is to attack the code from several angles to expose security threats

EXPERIMENT-1b / Project 7

Aim: Design of Test Suits for White Box Testing

Tools/Apparatus: None

Procedure:

1) Identify the feature, component, program to be tested.

2) Plot all possible paths in a flow graph.

3) Identify all possible paths from the flow graph.

4) Write Test Cases to cover every single path on the flow graph.

5) Execute, rinse, repeat.

Example:

White Box Mutation Testing: Mutation testing is often used to discover the best coding techniques to use for expanding a software solution.