Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

ACEDEMIC YEAR 2022-2023

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the

engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

By the end of the educational experience our students will be able to:

1. The Cyber Security graduates are able to gain a thorough understanding of the Cyber Security landscape with its growing threats and vulnerabilities in the world of computing including software and hardware.
2. Attain skills to comprehend and anticipate future challenges and devise methods to meet them and also, be articulate and skilled to convince all the stakeholders.
3. The Cyber Security graduates are able to acquire and demonstrate the ability to use ethical standard tools, practices and technologies for the analysis, design, development, implementation and testing of innovative and optimal Cyber Security solutions without compromising the privacy needs of individual and entities and the security concerns of law enforcement agencies.

Mapping of PSOs to POs:

PSO Number	PO Number
PSO1	PO1, PO2, PO6,
PSO2	PO4, PO9, PO10,
POS3	PO3, PO5, PO7, PO8, PO11 PO12

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Dr. Asha Durafe Program Coordinator Cyber Security Program

Lab Code	CSL501	**Lab Name**	Software Engineering Lab
Academic Year	2022-2023	**Semester**	V
Class	TE15	**Lab Coordinator**	Mrs.Rashmi Patel

Laboratory Outcomes (LO)

LO No.	LO Statement: On successful completion of laboratory experiments, learners will be able to :
1	Identify requirements and apply software process models to selected case study.
2	Develop architectural models for the selected case study.
3	Use computer-aided software engineering (CASE) tools.

List of Experiments

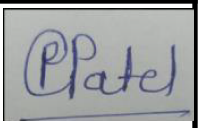
Sr. No.	Title	LO	PSO	PI
1	Application of at least two traditional process models.	1	3	3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.2.1
2	Application of the Agile process models.	1	3	3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.2.1
3	Preparation of software requirement specification (SRS) document in IEEE format.	2	3	3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.1.6,
4	Structured data flow analysis.	2	3	3.1.1, 3.1.2, 3.1.3, 3.1.5, 3.2.1, 3.2.2, 3.2.3
5	Use of metrics to estimate the cost.	2	3	11.1.1, 11.1.2, 11.2.1, 11.3.1
6	Scheduling & tracking of the project.	3	3	5.1.1, 5.1.2, 5.2.1,11.3.1, 11.3.2
7	Write test cases for black box testing.	3	3	7.1.1, 7.1.2, 7.2.1, 7.2.2
8	Write test cases for white box testing.	3	3	7.1.1, 7.1.2, 7.2.1, 7.2.2

9	Preparation of Risk Mitigation, Monitoring and Management Plan (RMMM).	3	3	7.1.1, 7.1.2, 7.2.1, 7.2.2
10	Version controlling of the project.	3	3	5.1.1, 5.1.2, 5.2.1, 5.2.2
11	Singleton Pattern	3	3	5.1.1, 5.1.2, 5.2.1, 5.2.2

Name: Mrs. Rashmi Patel Signature:

Date:

INDEX

Experiment Number: 1					
Date of Performance:		**01-08-2022**			
Date of Submission:		**08-08-2022**			
Program Execution/ formation/ correction/ ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
7	2	2	3	14	

Experiment No. 1

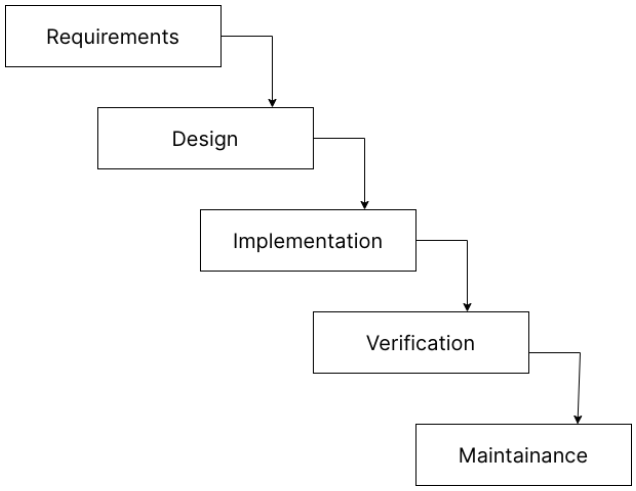
Aim: Application of at least two traditional process models.

Laboratory Outcome:CSL 501.1: Identify requirements and apply software process models to selected case study.

Problem Statement: Case study on Waterfall model & Spiral model using Bank System as an example.

Theory:

Waterfall process model:



1. **Requirements analysis and specification phase:** The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirements of the software. It describes the "what" of the system to be produced and not "how".In this phase, a large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language.

   In Bank System: Application Interface for Customers to Interact with the Banking app to access their Bank Account and provide Feedback. Bank admins will also be required to manage employee data. Good Security for transactions is a must. Database to store data of employees, customers, and

other data. Multi-Language Support.

2. **Design Phase:** This phase aims to transform the requirements gathered in the SRS into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a Software Design Document (SDD).

   In Bank System: Connection between Application and Database. Cryptographic Encryptions, MFA Service. Web Stack for Website and

   Application using programming languages and framework. Synchronized, Live transactions with minimum latency. Using Active Directory to manage Office PCs.

3. **Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD. During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

   In Bank System: Programming Language which is compatible cross platform (JavaScript, Flutter). In the testing Phase, check parameters for time complexity, space complexity, etc. Error Handling scripts and dependencies should be up to date.

4. **Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of

individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

In Bank System: Test for big data and capacity of server to handle requests. Penetration Testing and Bug testing must be performed to comply with security requirements and audits.

5. **Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

   In Bank System: Database, Active Directory Management. Regular security checks must be performed. Patches should be released regularly. Salary must be provided to developers who do these tasks regularly otherwise patches may contain viruses and banks may go bankrupt.

Some Circumstances where the use of the Waterfall model is most suited are:

- A project is short.
- The situation is calm.
- When the requirements are constant and not changed regularly.
- Where the tools and technology used are consistent.
- When resources are well prepared and are available to use.

Advantages of Waterfall model:

- This model is simple to implement and the number of resources that are

  required for it is minimal.
- The requirements are simple and explicitly declared; they remain unchanged during the entire project development.
- The start and end points for each phase are fixed, which makes it easy to cover progress.

- The release date for the complete product, as well as its final cost, can be determined before development.
- It gives easy control and clarity for the customer due to a strict reporting system.

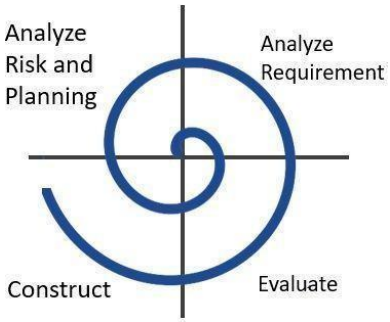
Disadvantages of Waterfall model:

- In this model, the risk factor is higher, so this model is not suitable for more significant and complex projects.
- This model cannot accept the changes in requirements during development.
- It becomes tough to go back to the phase. For example, if the application has now shifted to the coding phase, and there is a change in requirement, It becomes tough to go back and change it.
- Since the testing is done at a later stage, it does not allow identifying the challenges and risks in the earlier phase, so the risk reduction strategy is difficult to prepare.

Conclusion (Waterfall model):

- Oldest Software lifecycle model
- Used when requirements are well understood and risk is low.
- Workflow is in a linear (i.e, sequential) fashion.
- Used often with well-defined adaptations or enhancements to current software.
- Doesn’t support iteration, so changes can cause confusion.
- Difficult for customers to state all requirements explicitly and up front.
- Requires customer patience because a working version of the program doesn’t occur until the final phase.
- Problems can be somewhat alleviated in the model through the addition of feedback loops.

Spiral Model:



1. **Analyze Risk and Planning:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

In the Bank System: All the requirements from user perspective are taken care of and divided into phases with solutions. Each having alternative solutions and they are further analyzed.

2. **Analyze Requirement:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are

resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.

In Bank System: After the 1st phase we have a number of solutions for user requirements, now we have to combine them and make the best possible solution.

Once the solution is decided we have to look into risks associated with the solution. After this process we have design ready in our stake.

3. **Evaluate:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

In Bank System: In this phase will start implementation of the product. Testing and production will take place. And will gradually move to next production after completing small model of production

4. **Construct:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

In Bank System: It is time to take feedback from the user and implement new features which the user expected. And planning for the same will start in this phase.

Some Circumstances where the use of the Spiral Model is most suited are:

- When there is a budget constraint and risk evaluation is important.
- For medium to high-risk projects.
- Long-term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customers are not sure of their requirements which is usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

Advantages of Spiral Model:

1. **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
2. **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
3. **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
4. **Customer Satisfaction:** Customers can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

Disadvantages of Spiral Model:

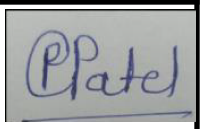
1. **Complex:** The Spiral Model is much more complex than other SDLC models.
2. **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
3. **Too much dependability on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced experts, it is going to be a failure to develop a project using this model.
4. **Difficulty in time management:** As the number of phases is unknown at the start of the project, so time estimation is very difficult.

Conclusion (Spiral model):

- The spiral model combines the idea of iterative development with

the systematic, controlled aspects of the waterfall model.

- This Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis.
- It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

Experiment Number: 2					
Date of Performance:		**08-08-2022**			
Date of Submission:		**22-08-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	2	12	

Experiment No. 2

Aim: Application of the Agile process models.

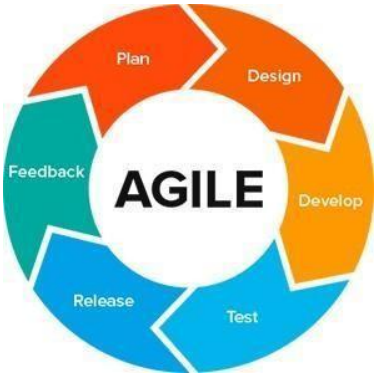
Laboratory Outcome: CSL 501.1: Identify requirements and apply software process models to selected case study.

Problem Statement: Case study on Agile model and its Application. Study SCRUM testing method of Agile model.

Theory:

Agile process model - The word Agile means swift or versatile. Agile process model refers to a software development approach based on iterative development. Agile methods break tasks into smaller

iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.



Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

Phases of Agile process model:

1. Requirements gathering (Plan)
2. Design the requirements (Design)

3. Construction/ iteration (Develop)
4. Testing/ Quality assurance (Test)
5. Deployment (Release)
6. Feedback

1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.

Time & Efforts will be dependent on Client need and Stakeholders expectations. There are lots of Software available for File Integrity Monitoring. All are the same with minimal unique features. To grow our software model for File Integrity

Monitoring we need a good marketing strategy to make brand awareness. Once the features of software is decided we can start with design
In File Integrity Monitor : Discussion with Developers to analyze requirements gathering about the project. Analyze costs of requirements like software, hardware, IT systems etc.

2. **Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.

In File Integrity Monitor: After gathering requirements, we start the design phase. Connection between Application and Database. Cryptographic Encryptions, MFA Service. Web Stack for Website and Application using programming languages and framework. Synchronized, Live logs with minimum latency. Using Active

Directory to manage Database.

3. **Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

In File Integrity Monitor: In this phase, we start actual implementation of the system. This includes developing the project using version control systems. This also includes managing the project workflow by a project lead.

4. **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

In Banking System: In the testing Phase, check parameters for time complexity, space complexity, etc. Error Handling scripts and dependencies should be up to date. Banking applications should also be easy to use for non-technical people. That's why in testing we also have to do GUI/UI tests by normal people.

5. **Deployment:** In this phase, the team issues a product for the user's work environment.

6. **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

In Banking System: In financial software it's very important to look after every feedback getting from customer, at the end it is all for the customer to feel safe and effortless while doing transactions. All the negative feedback should be improved as soon as possible.

Application of Agile Model:

1. When frequent changes are required.
2. When a highly qualified and experienced team is available.
3. When a customer is ready to have a meeting with a software team all the time.
4. When project size is small.

Advantage(Pros) of Agile Method:

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfills the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

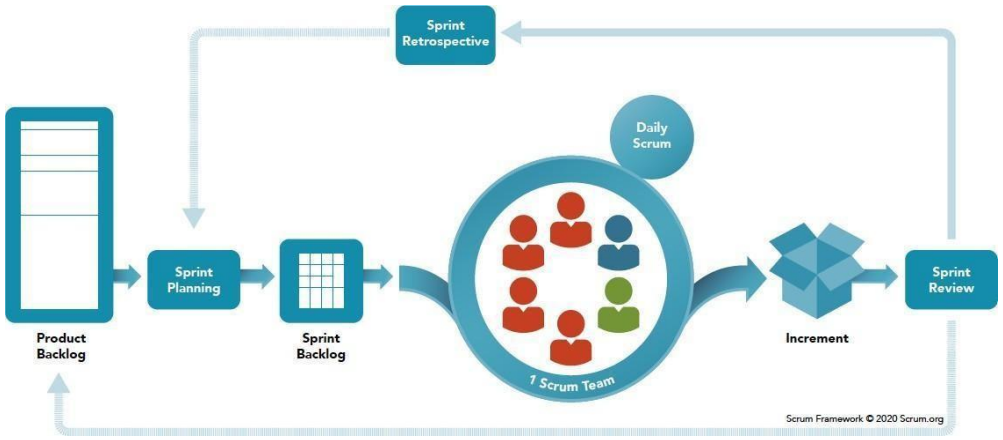
Disadvantages(Cons) of Agile Model:

1. Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
2. Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project

can become a difficulty.

Agile Testing Methods:

SCRUM is an agile development process focused primarily on ways to manage tasks in team-based development conditions.



Scrum in Software Testing is a methodology for building complex software applications. It provides easy solutions for executing complicated tasks. Scrum helps the development team to focus on all aspects of the software product development like quality, performance, usability and so on. It provides transparency, inspection and adaptation during the software development to avoid complexity.

Scrum Testing is a testing done in scrum methodology to verify the software application requirements are met. It involves checking non-functional parameters like security, usability, performance etc. There is no active role of tester in the process so it is usually performed by developers with Unit Test. Sometimes dedicated test teams are needed depending on the nature & complexity of the project.

There are three roles in it, and their responsibilities are:

- Scrum Master: The scrum can set up the master team, arrange the meeting and remove obstacles for the process
- Product owner: The product owner makes the product backlog, prioritizes the delay and is responsible for the distribution of functionality on each repetition.
- Scrum Team: The team manages its work and organizes the work to complete the sprint or cycle.

Key features of Scrum Testing Model:

1. Scrum has a short fixed schedule of release cycles with adjustable scope known as sprints to address rapidly changing development needs. Each release could have multiple sprints. Each Scrum Project could have multiple Release Cycles.
2. A repeating sequence of meetings, events, and milestones
3. A practice of testing and implementing new requirements, known as stories, to make sure some work is released ready after each sprint

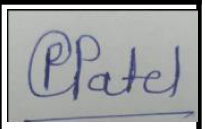
Conclusion: In this experiment, we did a case study on Agile process models on the “File Integrity Monitor” as our case study

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 3					
Date of Performance:	**08-08-2022**				
Date of Submission:	**22-08-2022**				
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	3	13	

Experiment No. 3

Aim: Preparation of software requirement specification (SRS) document in IEEE format.

Laboratory Outcome: CSL 501.2: Develop architectural models for the selected case study.

Problem Statement: Prepare an SRS Document for an File Integrity Monitor

Documentation:

Table Of Contents

1.Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the open-source software FIM. This document will explain features, use case software requirements of the software. This document is aimed at users as well as developers of this software.

1.2 Aimed audience

- Developers who are interested in developing the software and making it better, add more user convenience features, and fix bugs.
- Typical users, Home System User who want to use FIM and monitor their folders.

- Advanced users such as System Admin, System Analyst to maintain security of system.
- Security team of a organization can use to analyse the attack if performed on the system.

1.3 Product Scope

FIM stands for File Integrity Monitoring. Using this tool user can monitor the folders where sensitive, confidential, important data is saved. They can also see the results in form of graphs so it becomes easy to analyse as admin of a system. FIM aims at maintaining Integrity of Information which is a basic element of information security.

1. Overall Description

1.1 Product Perspective

Gephi is developed for people who are security conscious and want to monitor all of there important files and folders and also organisations who want to analyse there systems and make sure that integrity of their data is maintained. The tool will show all types of modifications that can be done with a file like creation, deletion, rename etc. The product also includes graphs which will make it easy for users to analyse the logs. It is an open source project which aims at maintaining integrity of data. It is developed to run on Linux and Windows.

1.2 Product Functions

1.3 User Classes

- Developers who are interested in developing the software and making it better, add more user convenience features, and fix bugs.
- Typical users, Home System User who want to use FIM and monitor their folders.
- Advanced users such as System Admin, System Analyst to maintain security of system.
- Security team of a organization can use to analyse the attack if performed on the system.

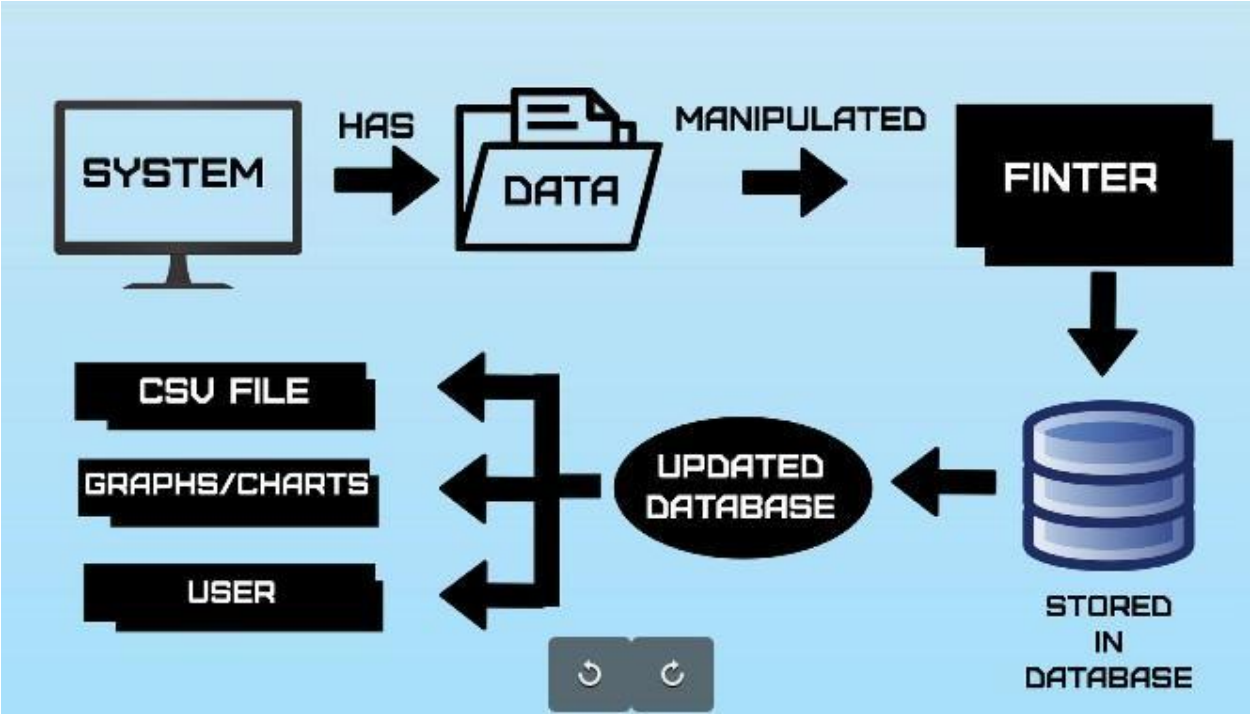
1.4 Operating Environment

- Linux
- Windows 11
- Windows 10
- Windows 8

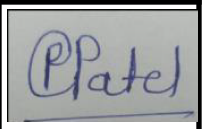
1.5 Design and Implementation Constraints

FIM is developed in python and used kivy for GUI. Since it is developed in python it is system independent.

2.Data Flow Diagram



Conclusion: In this experiment, we prepared an SRS Document for an File Integrity Monitor.

Experiment Number: 4					
Date of Performance:		**22-08-2022**			
Date of Submission:		**29-08-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	2	12	

Experiment No. 4

Aim: Structured data flow analysis.

Laboratory Outcome: CSL 501.2: Develop architectural models for the selected case study.

Problem Statement: Perform a Structured data flow analysis diagram on any topic in Software Engineering.

Related Theory:

Structured Analysis is a development method that allows the analyst to understand the system and its activities in a logical way.

It is a systematic approach, which uses graphical tools that analyze and refine the objectives of an existing system and develop a new system specification which

can be easily understandable by user.

It has following attributes −

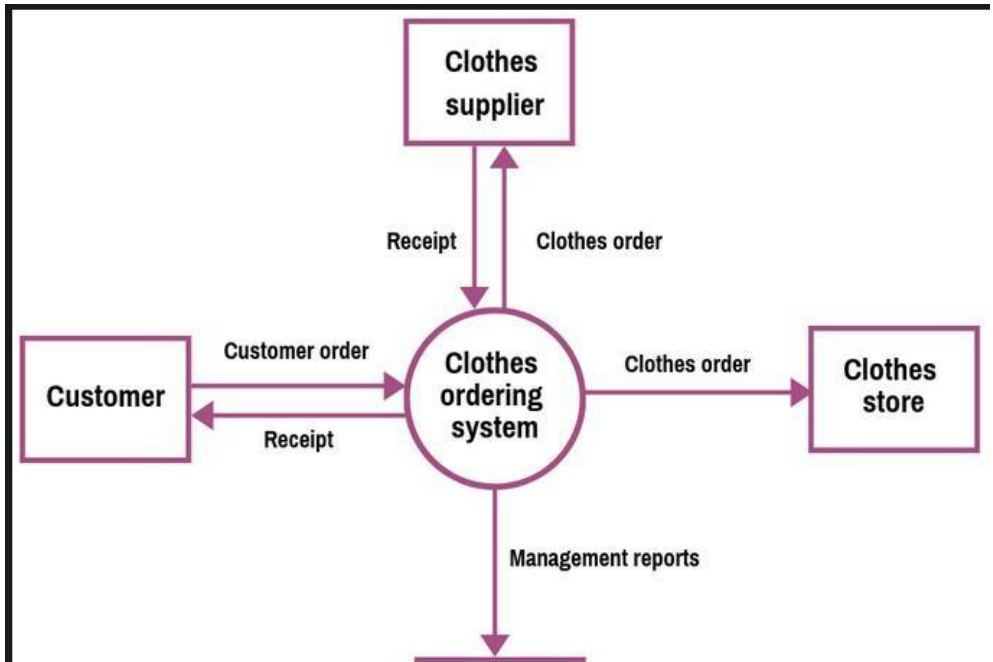
1. It is graphic which specifies the presentation of application.
2. It divides the processes so that it gives a clear picture of system flow.
3. It is logical rather than physical i.e., the elements of system do not depend on vendor or hardware.
4. It is an approach that works from high-level overviews to lower-level details.

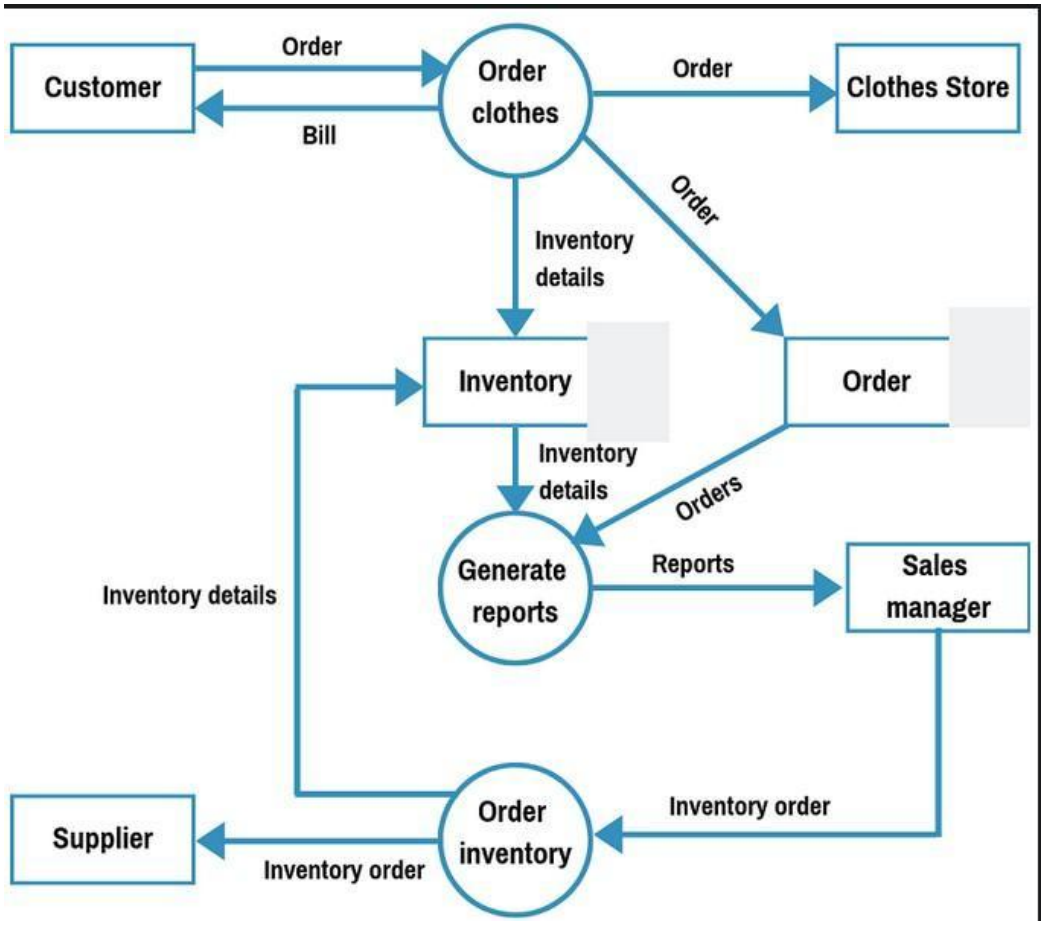
During Structured Analysis, various tools and techniques are used for system development.

- Data Flow Diagrams
- Data Dictionary
- Decision Trees
- Decision Tables
- Structured English
- Pseudocode

Program Listing And Output:

Cloth Ordering System Level 0 Diagram:





Level 1 diagram

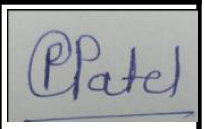
Conclusion: In this experiment, we studied the structured data flow analysis of a Cloth Ordering System.

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

Experiment Number: 5					
Date of Performance:		22-08-2022			
Date of Submission:		29-08-2022			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	3	13	

Experiment No. 5

Aim: Use of metrics to estimate the cost.

Laboratory Outcome: CSL 501.2: Develop architectural models for the selected case study.

Problem Statement: Develop a document to estimate the cost of a Library Management System.

Related Theory:

For any new software project, it is necessary to know how much it will cost to develop and how much development time it will take. Several estimation procedures have been developed and have the following attributes in common.

1. Project scope must be established in advance.

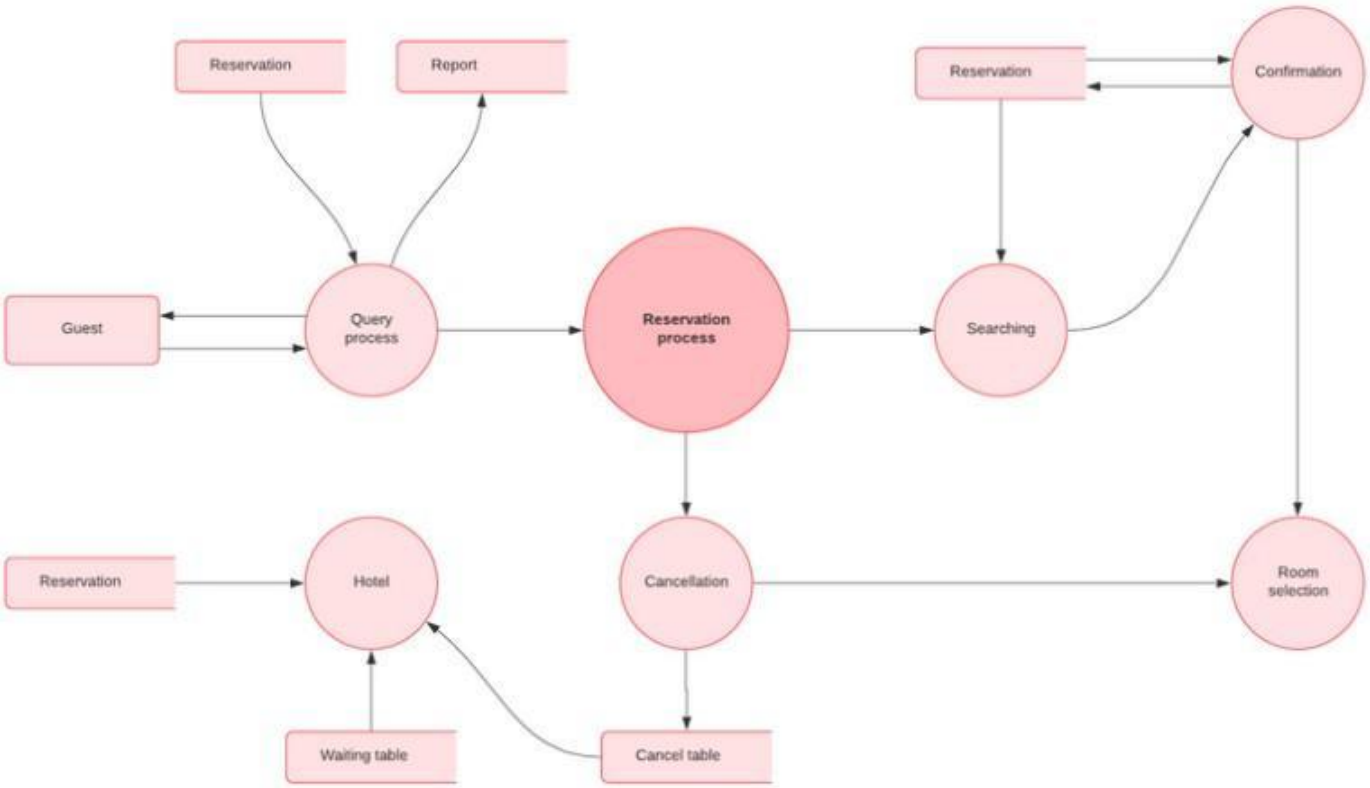
2. Software metrics are used as a support from which evaluation is made.
3. The project is broken into small PCs which are estimated individually.
4. To achieve true cost & schedule estimate, several options arise.
5. Delay estimation

1. Used symbol decomposition techniques to generate project cost and schedule estimates.
2. Acquire one or more automated estimation tools.

During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule. In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.

Program Listing And Output:

Step 1: Draw DFD level 2



Step 2: Formula of FP

$$FP= \text{Count Total} * [\ 0.65 + 0.01 * \Sigma\ (Fi)\ ]$$

Step 3:

Measurement Parameter	count		Weighing factor			
			simple	average	complex	
No of EI	5	*	3	4	6 =	15
No of EO	6	*	4	5	7 =	24
No of EQ	5	*	3	4	6 =	15
No of ILF	5	*	7	10	15 =	35
No of EIF	5	*	5	7	10=	25
Count-total						114

Assume :-

EI=5,EO=6,EQ=5,ILF=5,EIF=5

Step 4: Justify your view why you have chosen your application as simple or average or complex.

Q. No.	Questions	Rating
1	Does the system require reliable backup and recovery?	3
2	Are specialized data communications required to transfer information to or from the application?	4
3	Are there distributed processing functions?	4
4	Is performance critical?	1
5	Will the system run in an existing, heavily utilized operational environment?	3
6	Does the system require online data entry?	5

7	Does the online data entry require the input transaction to be built over multiple screens or operations?	5
8	Are the ILFs updated online?	4
9	Are the inputs, outputs, files, or inquiries complex?	2
10	Is the internal processing complex?	5
11	Is the code designed to be reusable?	3

12	Are conversion and installation included in the design?	5
13	Is the system designed for multiple installations in different organizations?	4
14	Is the application designed to facilitate change and ease of use by the user?	2

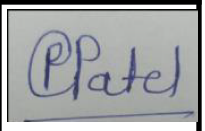
Σ (Fi) = 50

Count Total =
114 Σ (Fi) = 50

Therefore ,
FP = Count Total * [0.65 + 0.01 * Σ (Fi)]
FP=114*[0.65+0.01*50]
FP=131.1

The Function Point Of System is 131.1

Conclusion: We successfully performed cost estimation using metrics.

Experiment Number: 6					
Date of Performance:	**29-08-2022**				
Date of Submission:	**05-09-2022**				
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	2	3	13	

Aim: Scheduling & tracking of the project.

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: Perform Scheduling and Tracking of the project with the help of Gantt Chart.

Related Theory:

Scheduling

- Task networks (activity networks) are graphic representations can be of the task interdependencies and can help define a rough

schedule for particular project

- Scheduling tools should be used to schedule any non-trivial project.
- Program evaluation and review technique (PERT) and critical path method (CPM) are quantitative techniques that allow software planners to identify the chain of dependent

tasks in the project work breakdown structure (WBS) that determine the project duration time.

- Timeline (Gantt) charts enable software planners to determine what tasks will be need to be conducted at a given point in time (based on estimates for effort, start time, and duration for each task).
- The best indicator of progress is the completion and successful review of a defined software work product.
- Time-boxing is the practice of deciding a priori the fixed amount of time that can be spent on each task. When the task's time limit is exceeded, development moves on to the next task (with the hope that a majority of the critical work was completed before time ran out).

Tracking Project Schedules

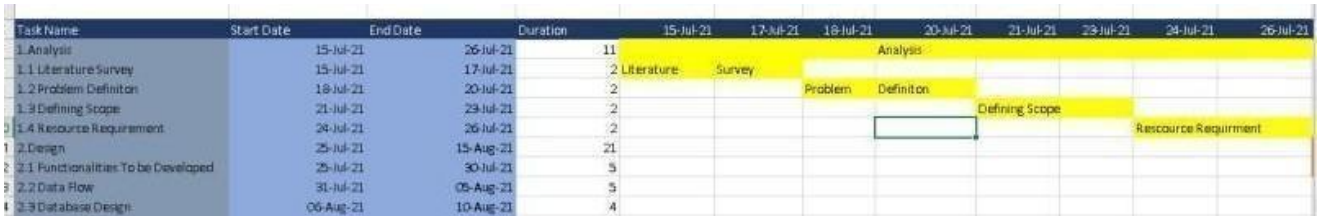
- Periodic project status meetings with each team member reporting progress and problems
- Evaluation of results of all work product reviews
- Comparing actual milestone completion dates to scheduled dates
- Comparing actual project task start-dates to scheduled start-dates
- Informal meeting with practitioners to have them asses subjectively progress to date and future problems

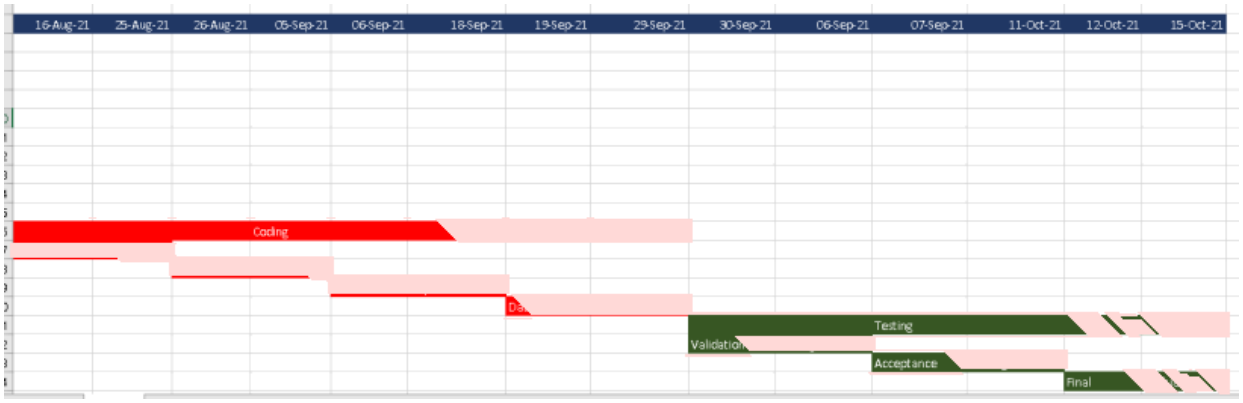
Use earned value analysis to assess progress quantitatively

Program Listing & Output:

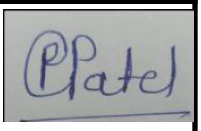
Add tasks and details by referencing the Software Development Life Cycle Model.

Case Study on Online Banking System





Conclusion: In this experiment, we implemented a project's scheduling and tracking using Gantt Chart.

Experiment Number: 7					
Date of Performance:	**29-08-2022**				
Date of Submission:	**05-09-2022**				
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
7	2	2	3	14	

Aim: Write test cases for black box testing.

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: Implement the scenario for Blackbox testing using CASE tools.

Related Theory:

Black box testing is a type of software testing in which the functionality of the software is not known. The testing is done without the internal knowledge of the products.

Black box testing can be done in the following ways:

1. Syntax Driven Testing – This type of testing is applied to systems that can be syntactically represented by some language. For example- compilers, language

that can be represented by context-free grammar. In this, the test cases are generated so that each grammar rule is used at least once.

2. Equivalence partitioning – It is often seen that many types of inputs work similarly so instead of giving all of them separately we can group them and test only one input of each group. The idea is to partition the input domain of the system into several equivalence classes such that each member of the class works similarly, i.e., if a test case in one class results in some error, other members of the class would also result in the same error.

3. Boundary value analysis – Boundaries are very good places for errors to occur. Hence if test cases are designed for boundary values of the input domain then the efficiency of testing improves and the probability of finding errors also increases. For example – If the valid range is 10 to 100 then test for 10,100 also apart from valid and invalid inputs.

4. Cause effect Graphing – This technique establishes a relationship between logical input called causes with corresponding actions called the effect. The causes and effects are represented using Boolean graphs.

5. Requirement-based testing – It includes validating the requirements given in the SRS of a software system.

6. Compatibility testing – The test case result not only depends on the product but is also on the infrastructure for delivering functionality. When the infrastructure parameters are changed it is still expected to work properly.

Program Listing And Output:

Test Case 1: Admin Login

SR. NO	TEST CASES	EXPECTED RESULT	TEST RESULT
1	Enter valid username and password and click on login button	Software should display main window	Successful
2	Enter invalid credentials	Software should not display main window	Successful

Test Case 2: Create File in System

SR. NO	TEST CASES	EXPECTED RESULT	TEST RESULT
1	On creation of file	Creation message in displayed	Successful

2	On deletion of file	Deletion message in displayed	Successful
3	On modification on file	Modified msg is displayed	Successful

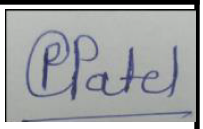
Test Case 3: Database Update

SR. NO	TEST CASES	EXPECTED RESULT	TEST RESULT
1	On creation of file	Updates DB with event, path, date and time	Successful
2	On deletion of file	Updates DB with event, path, date and time	Successful

Test Case 4: Modify Graph

SR. NO	TEST CASES	EXPECTED RESULT	TEST RESULT
1	DB update	Updates Pie chart	Failed
2	DB update	Updates Line Chart	Successful
3	DB Update	Updates Counter	Successful

Conclusion: In this experiment, we successfully implemented Black Box Testing of an File Integrity Monitor.

Experiment Number: 8					
Date of Performance:		**05-09-2022**			
Date of Submission:		**23-09-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
7	2	2	3	14	

Experiment No. 8

Aim: Write test cases for white box testing.

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: Implement the scenario for Whitebox testing using CASE tools.

Related Theory:

White box testing techniques analyze the internal structures, the used data structures, internal design, code structure and the working of the software rather than just the functionality as in black box testing. It is also called glass box testing or clear box testing or structural testing.

Working process of white box testing:

1. Input: Requirements, Functional specifications, design documents, source code.
2. Processing: Performing risk analysis for guiding through the entire process.
3. Proper test planning: Designing test cases so as to cover the entire code. Execute rinse-repeat until error-free software is reached. Also, the results are communicated.
4. Output: Preparing final report of the entire testing process.

Advantages:

- White box testing is very thorough as the entire code and structures are tested.
- It results in the optimization of code removing error and helps in removing extra lines of code.
- It can start at an earlier stage as it doesn't require any interface as in case of black box testing.

Disadvantages:

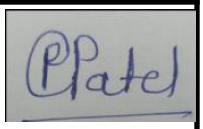
- Main disadvantage is that it is very expensive.
- Redesign of code and rewriting code needs test cases to be written again.
- Testers are required to have in-depth knowledge of the code and programming language as opposed to black box testing.
- Missing functionalities cannot be detected as the code that exists is tested.

Program Listing And Output:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Result	Pass/ Fail
TC01	Check admin login with valid data	Go to the website and enter UserID and Password. Then, click login	UserID and Password is "admin"	Logged in Dashboard	As expected	Pass
TC02	Test the input values of different events	Go to the directory and create/del/modify/move a file	Make some modification in selected directory	Return the event occured	As Expected	Pass

TC03	Test the input values updating the database	Go to the directory and create/del/modify/move a file	Make some modification in selected directory	Tables in DB gets updated	As Expected	Pass
TC04	Test the input values for charts updating	Do some event	Modified DB	Charts get updated	No live update chart	Fail

Conclusion: In this experiment, we implemented whitebox testing of the File Integrity Monitor

Experiment Number: 9					
Date of Performance:		**05-09-2022**			
Date of Submission:		**23-09-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
7	2	3	3	15	

Experiment No. 9

Aim: Preparation of Risk Mitigation, Monitoring and Management Plan (RMMM).

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: Prepare a document that demonstrates the RMMM plan.

Related Theory:

A risk management technique is usually seen in the software Project plan. This can be divided into Risk Mitigation, Monitoring, and Management Plan (RMMM). In this plan, all work is done as part of risk analysis. As part of the overall project plan, the project manager generally uses this RMMM plan.

In some software teams, risk is documented with the help of a Risk Information

Sheet (RIS). This RIS is controlled by using a database system for easier management of information i.e creation, priority ordering, searching, and other analysis. After documentation of RMMM and start of a project, risk mitigation and monitoring steps will start.

Risk Mitigation :
It is an activity used to avoid problems (Risk Avoidance).
Steps for mitigating the risks as follows.

- Finding out the risk.
- Removing causes that are the reason for risk creation.
- Controlling the corresponding documents from time to time.
- Conducting timely reviews to speed up the work.

Risk Monitoring :
It is an activity used for project tracking.
It has the following primary objectives as follows.

- To check if predicted risks occur or not.
- To ensure proper application of risk aversion steps defined for risk.
- To collect data for future risk analysis.
- To allocate what problems are caused by which risks throughout the project.

Risk Management and planning :
It assumes that the mitigation activity failed and the risk is a reality. This task is done by Project manager when risk becomes reality and causes severe problems. If the project manager effectively uses project mitigation to remove risks successfully then it is easier to manage the risks. This shows the response that will be taken for each risk by a manager. The main objective of the risk management plan is the risk register. This risk register describes and focuses on the predicted threats to a software project

Program Listing And Output:

RISK INFORMATION SHEET			
RISK ID: R-01	DATE: 20/10/2021	PROBABILITY: 40%	IMPACT: CATASTROPHIC
RISK DESCRIPTION: Member Risk: The risk will cause due to unskilled group members in the group			
REFINEMENT/CONTEXT: Sub-condition 1: If the team members are not experienced enough to use the application necessary to develop the software it will keep pushing the development dates until it's too late to save the project. Sub-condition 2: If one or more members of the software development team are not putting in all the effort required to finish the project it will cause the project to fail. .			
MITIGATION/MONITORING: This risk concerns the knowledge of the employees and their willingness to help make the project succeed. As a mitigation step of this risk we will make sure that someone in all of the project development phases knows exactly what to do and the tools to use to achieve the goals. If the employees that have little knowledge in the main software implementation language fail to learn it, it may cause big problems when the coding part begins.			
MANAGEMENT / CONTINGENCY PLAN / TRIGGER: Monitoring and managing of this risk will include looking out for each other, that is if some team-member is having difficulties in performing some tasks or using a particular tool or technique other members of that team will help him out. This is where team members may have to spend little time learning or teaching what others know. If team member lacks ability to use certain programming language or application, other team members will take some time off to teach the team member basics related to that application.			
ORIGINATOR: Dante Zogratis Richita			
ASSIGNED TO: Asta Staria			

Mahavir Education Trust's

SHAH & ANCHOR KUTCHHI ENGINEERING COLLEGE

Chembur, Mumbai - 400 088

UG Program in Cyber Security

RISK INFORMATION SHEET			
RISK ID: R-02	DATE: 22/10/2021	PROBABILITY: 35%	IMPACT: CATASTROPHIC

RISK DESCRIPTION:

Process risk involves risks regarding product quality. If the product developed does not meet the standards set by the customer or the development team is a failure.

REFINEMENT/CONTEXT:
Sub-condition 1: This can happen because of the customer's failure to describe the true business need or the failure of the software development team to understand the project and than to use proper equipment and employees to finish the project.

MITIGATION/MONITORING:

We want the quality of the product to be as high as possible. To achieve this we will set up guidelines to be followed for each of the team members during all the phases of the software development cycle. The standard will be set and defined for all of the software development. This will help the team in delivering the high quality product thus increasing our reputation in the market. This will help bring in more clients in the future. It will also save customers from getting low quality products.

MANAGEMENT / CONTINGENCY PLAN / TRIGGER:

To monitor the risk here we will review each other's work to find the problems and to help each other in achieving better product quality. We will also have the general guidelines set for all of the work to be carried on for the software development. Software development team will constantly check each others work; will compare it with the set guidelines, and will inform a team member who is failing to participate in following the guidelines.

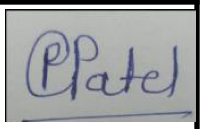
ORIGINATOR:

Minato

ASSIGNED TO:

Kakashi

Conclusion: In this Experiment, we prepared a document that demonstrates the RMMM plan.

Experiment Number: 10					
Date of Performance:		**23-09-2022**			
Date of Submission:		**30-10-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	3	3	14	

Experiment No. 10

Aim: Version controlling of the project.

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: Implement Version Controlling of a sample project using CASE tools.

Related Theory:

What is a “version control system”?

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.

Why is the Version Control system so Important?

As we know that a software product is developed in collaboration by a group of developers they might be located at different locations and each one of them contributes to some specific kind of functionality/features. So in order to contribute to the product, they made modifications to the source code (either by adding or removing).

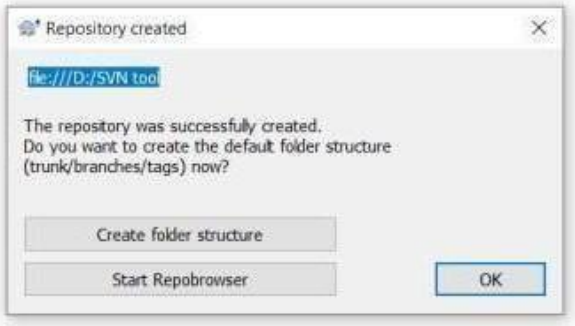
A version control system is a kind of software that helps the developer team to efficiently communicate and manage (track) all the changes that have been made to the source code along with the information like who made and what changes have been made.

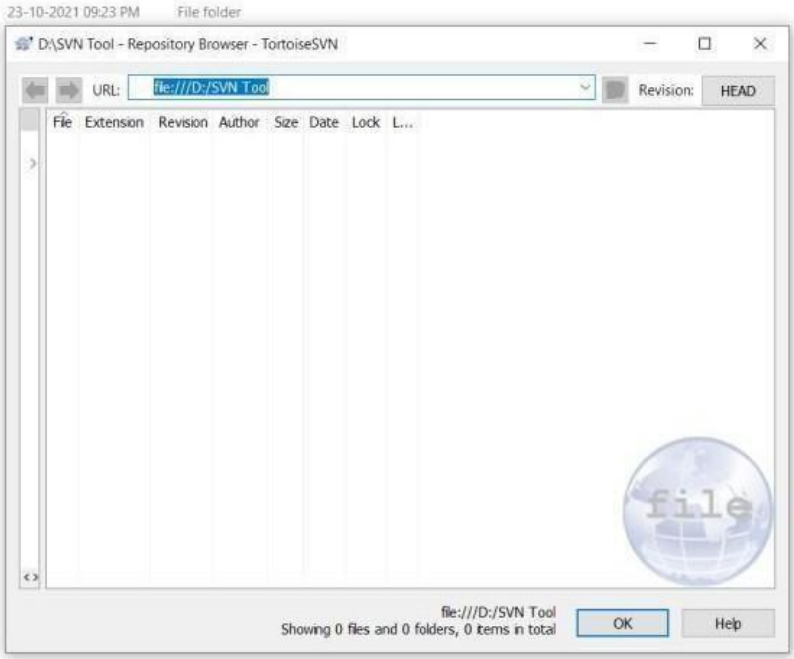
A separate branch is created for every contributor who made the changes and the changes aren't merged into the original source code unless all are analyzed. As soon as the changes are green-signaled they merge to the main source code. It not only keeps source code organized but also improves productivity by making the development process smooth.

Program Listing And Output:

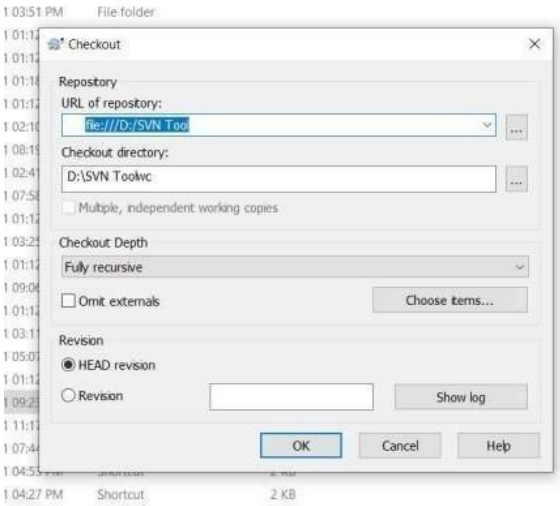
Installation of SVN – Tortoise

SVN repository created

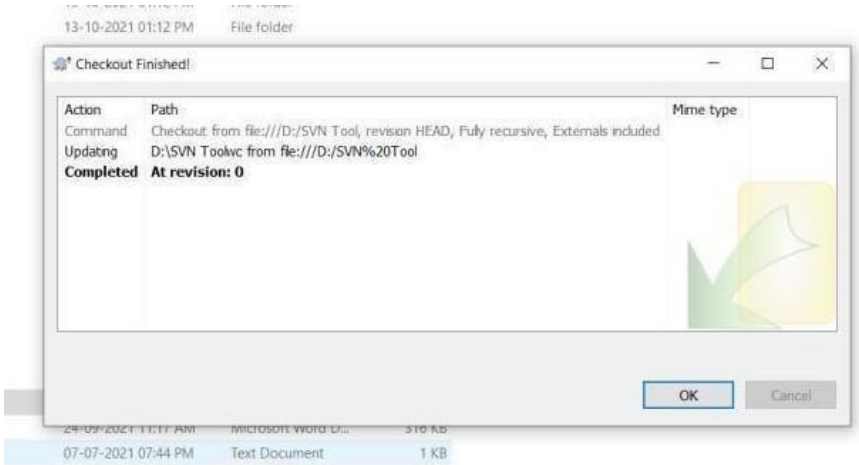




Checkout Dialogue Box: Browsing the repository.

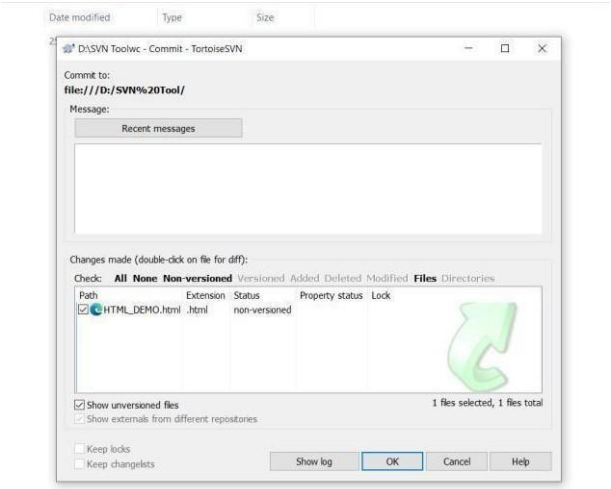


Checkout Finished

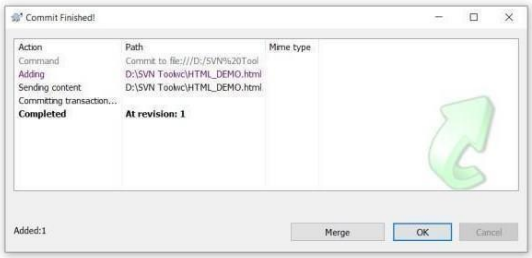


Creating HTML

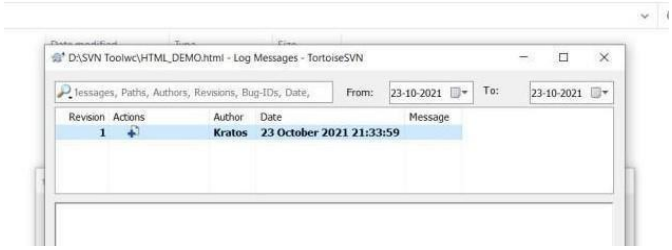
Document File Commit



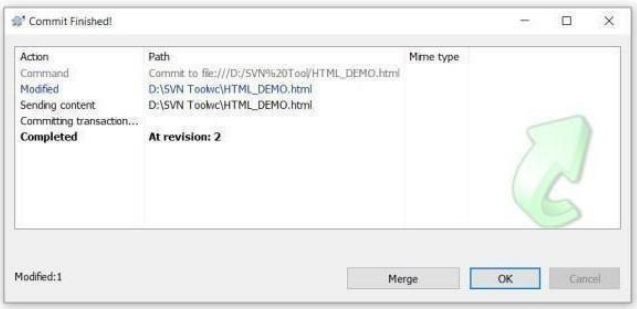
Commit Finished



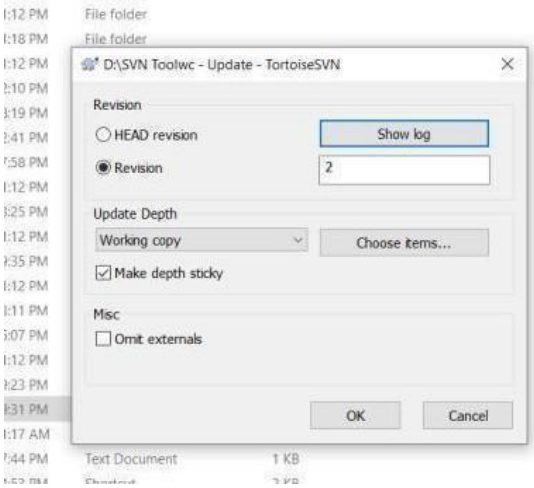
SVN File Update



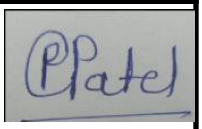
File Commit Finished



Commit Logs



Conclusion: In this experiment, we successfully demonstrated the Implementation of Version Control System.

Experiment Number: 11					
Date of Performance:		**30-09-2022**			
Date of Submission:		**07-10-2022**			
Program Execution/ formation/ correction / ethical practices (07)	**Documentation (02)**	**Timely Submission (03)**	**Viva Answer to sample questions (03)**	**Experiment Total (15)**	**Sign**
6	2	3	3	14	

Experiment No. 11

Aim: Implement Singleton Pattern

Laboratory Outcome: CSL 501.3: Use computer-aided software engineering (CASE) tools.

Problem Statement: To study design pattern and implement Singleton pattern.

Related Theory:

Design Pattern:

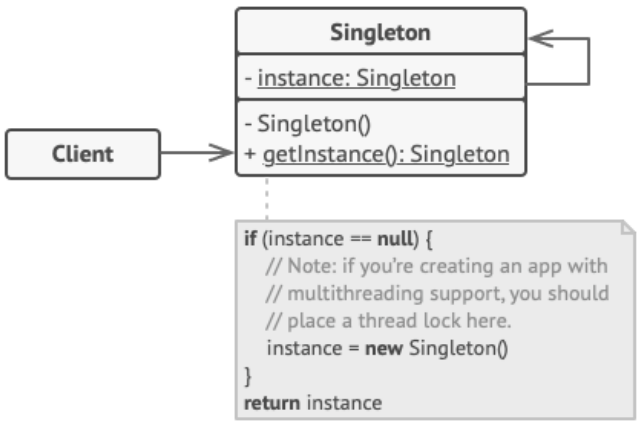
- Design patterns represent the best practices used by experienced object-oriented software developers.

- Design patterns are solutions to common issues that software developers face when developing software.
- Several software engineers used trial and error to arrive at these solutions over a long period of time.

Singleton Pattern:

- The singleton pattern is a design pattern that restricts the instantiation of a class to one object.
- Singleton pattern is one of the simplest design patterns.
- This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.
- This pattern involves a single class which is responsible for creating an object while making sure that only a single object gets created.

Program Listing And Output:




```
class SingletonClass(object):
   def __new__(cls):
       if not hasattr(cls, 'instance'):
           cls.instance = super(SingletonClass, cls).__new__(cls)
       return cls.instance


singleton = SingletonClass()
new_singleton = SingletonClass()

print(singleton is new_singleton)
```

Output:

Conclusion: Here, we studied the design patterns and implemented the Singleton pattern.

Assignment – 1

Marks Obtained – 18/20

Assignment – 2

Marks Obtained – 14/20