

Experiment 1

SOLUTION: Study the complete Software Development Life Cycle (SDLC) and analyze various activities conducted as a part of various phases of SDLC of “ Book Store Management Application ”.

Analysis :-

1. Planning and Requirement Analysis

Objective: -

The main aim of this phase is to do feasibility analysis of the product on the basis of inputs provided by client, market surveys and domain experts in the industry.

1) Client will provide inputs:-

- a) Information about books including price , author , edition , etc.
- b) Budget available for project.
- c) Project Deadline.
- d) Real time updated information.

1) Market survey input:-

- a) Is there is any other product available in the market?
- b) Is budget is sufficient or not to provide service?
- c) What are the challenges in creating a book store management application?
- d) Risk percentage analysis.

2) Technical team inputs:-

- a) Does the team have technical skills to develop the products?
- b) Can they meet the deadline?
- c) Which model they will use?

Outcome: -

We got all the client requirements. We have got the inputs that we can complete the product within deadline within budget mentioned by our client and we can move ahead. On the basis of market survey inputs and technical team inputs we can now move ahead to the next phase.

2. Defining product requirements

Objective: In the first phase we were defining the client requirements but now we will define the product requirement formally in Software Requirement Specification (SRS) document.

It includes following specification: -

Cloud server or data warehouse needed to store application data.

Link to publishers for books related data.

Outcome:-

After this phase we have clearly defined our product requirements and documented it in SRS document. Now we can move further to next phase.

3. Designing the product architecture **Objective:**

In this phase we will design a complete framework for our product. We will make things ready so that technical or developer team can work upon them in next phase. We will set up things according to our SRS document.

Processes done in this phase:

Defining product's rough framework.

Setting up data warehouse.

Contract with publishers and authors.

Outcome:

We have successfully deployed the architecture needed for the next phase of SDLC and now we can move to coding phase of SDLC.

4. .Building and developing product

Objective:-

This phase includes the actual implementation of the project. This phase includes coding phase of SDLC. It includes mainly the technical teams of industry. In this phase a proper working book store management application will be deployed.

Data warehouse will be setup with the application.

Outcomes:-

After this phase of SDLC we have deployed proper working product with all the services mentioned by client in phase 1.

5. Testing

Objective:

It is one the most important phase of SDLC. Making a robust application is very important in the industry as users trusts only in reliable and

robust products. Any loop hole in the product can create a big problem for the industry, client and user satisfaction is very important. It includes following processes proper testing against test sets. Increasing the accuracy in predictions. Reduce risk of errors and failures.

Outcomes:

After this phase our Book store management application is fully ready to be deployed in the market.

We can now finally provide service to the client.

6. Maintenance Objective:

The last stage of the SDLC cycle is the maintenance. There are some cases where a industry or organization have to look upon the software even after handing it to the client till the date it provides services to the client. In those scenarios maintenance of the software is taken care by the industry.

Following process is one in this phase:

Life time assistance is provided by the industry.

Parallel maintenance and development is required in the software and after all the

changes the product is released with new version and deployed parallel to all the systems.

If client requirements change then again this is handled by the industry.

Outcomes:

The entire software is built and deployed. Now we are done with our software \and we will provide life time maintenance and assistance to the software

Experiment 2**Software Crisis Introduction:**

Software crisis is the situation resulted due to the catastrophic failure of software development which leads to incomplete and degrading performance of software products.

- Term was coined in the year 1968.
- In general it refers to poorly written, hard to read, error-prone software that often lacks good documentation.

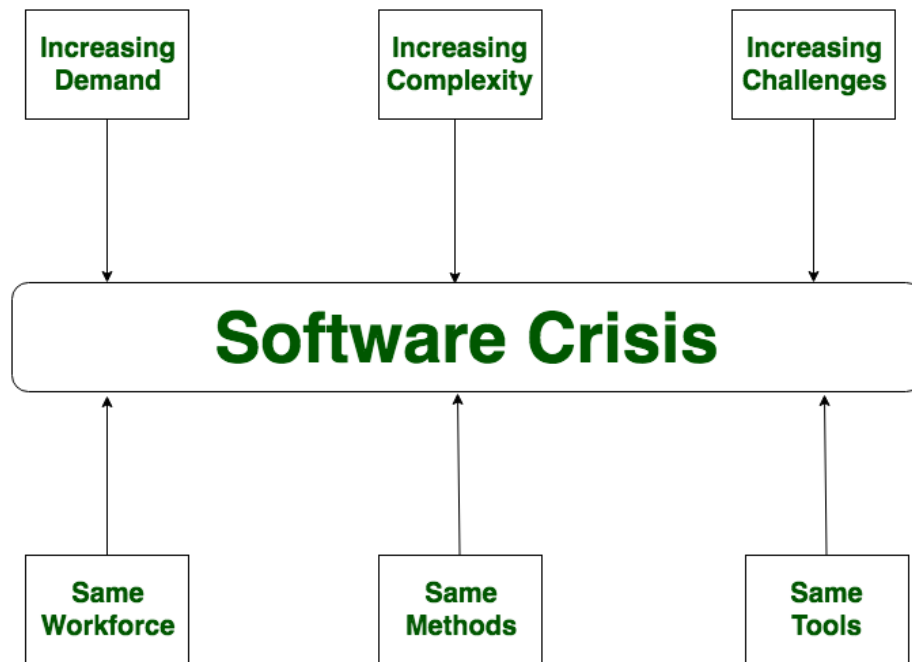
The term software crisis revolves around three concepts: complexity, change and the expectations.

Software Crisis in terms of statistics in 1990's

- * 31 % of projects canceled
- * 52.7% cost an average of 189% over budget
- * 84% are late or over budget (91% for large companies.)
- * The average system is delivered without 58% of proposed functionalities
- * \$81 billion in 1995 for cancelled projects
- * \$51 billion in 1995 for over-budget projects

Only 16.2% of software projects are completed on-time and on-budget. In larger companies, a meager 9% of technology projects come in on-time and on-budget.

The causes of software crisis were linked to the overall complexity of the software process and the relative immaturity of software engineering as a profession. The main reason for the crisis is the lack of a sound software construction methodology with which to manage the high complexity of modern applications.



Causes of software crisis:

- 1) Due to the Projects running over-budget
- 2) Due to the Projects running over-time
- 3) Software was very inefficient and was of low quality
- 4) Software often did not meet requirements
- 5) code was difficult to maintain and Projects were a mess and unmanageable.

solutions for preventing software crisis:-

One of the possible solutions to the software crisis is the study of software engineering.

- Experience working as a team member on a software development project.
- Knowledge of basic statics and experimental design.

□ using tools that help us manage this complexity.

- Less time and fewer people needed for productive innovation.

Experiment 2 (Software Crisis)

Objective To identify the problem related to software crisis for a given scenario

Background In the early years of computers applications, the focus of the development and innovation were on hardware. Software was largely views as an afterthought. Computer Programming was an art. Programmers did not follow any disciplined or formalized approaches.

This way of doing things was adequate for a while, until the sophisticated of computer applications outgrow. Software soon took over and more functions which were done manually. A software houses begin to develop for widespread distribution. Software development projects produced thousands of source program statement. With the increase in the size and complexity of the software, following situation resulted is collectively termed as software crisis.

1. Time Slippage
2. Cost Slippage
3. Failure at customer Site
4. Intractable Error after delivery

ProblemDescription

In the context of this background, for each of the scenario mentioned below, identify the most appropriate problem related to software crisis and mention the same in the table provided.

Scenario A

Railways reservation software was delivered to the customer and was installed in one of the metro station at 12.00 AM (mid-night) as per the plan. The system worked quite fine till the next day 12.00 PM(noon). The system crashed at

12.00 PM and the railways authorities could not continue using software for reservation till 02.00 M. It took two hours to fix the defect in the software in the software.

Scenario B

A polar satellite launch vehicle was scheduled for the launch on August 15th. The auto-pilot of the rocket to be delivered for integration of the rocket on may 15th. The design and development of the software for the auto-pilot more effort because of which the auto-pilot was delivered for the integration on June 15th (delayed by a month). The rocket was launched on Sep 15th (delayed by a month).

Scenario C

Software for financial systems was delivered to the customer. Customer informed the development team about a mal-function in the system. As the software was huge and complex, the development team could not identify the defect in the software.

Scenario D

Due to the defect in the software for the baggage handling system. There was also of & 2M of revenues for the airport authorities.

Scenario Situation (as given A to D)

Scenario	Situation
A	Intractable Error after delivery
B	Time Slippage
C	Failure at Customer Site
D	Cost Slippage

Practical 3

Objective: To identify the suitable software development model for the given case studies/scenario.

Problem description:

Identify the most appropriate SDLC model for these scenario.

Case Study-1

“ABC” is a new enterprise recently started by a few young entrepreneurs, intended to develop robotic systems using IT based solutions. “TENZ” is a well-established automobile manufacturing plant, interested in automating some of the operations of the plant as first step towards automation, TENZ has decided to deploy a robotic arm to pick and place large number of bolts from one tray of a conveyer belt to another tray of a different conveyer belt. Moving one bolt at a time. TENZ has asked ABC for demonstration the robotic arm for the said purpose as a proof of concept. This is the first project being executed by ABC. TENZ would like to have a first experience of the robotic arm before signing the MOU (Memorandum of Understanding) with ABC. After MOU sign-off, TENZ would provide the complete set of requirement of the robot.

Case Study-2

ABC is an international software house. ABC is currently working on a project that is totally new for the development team and even the client is confused about the requirements of this project. Hence this company is facing difficulties because they fail to apprehend user requirements properly. For this project, it is decided to build a sample application and show it to the client for feedback. In the context of this above scenario as a project manager what will be the choice of the software lifecycle model?

Case Study-3

SMART_TECHY is a software development company working for online shop of electronic equipment. The different tasks that are expected from the teams in the company are as follows:

- The registered users must be allotted with an initial password after registration the user should be able to change the password keep his/her account secure.

- The people who are browsing through the shopping website, want to subscribe through the mailing list so that they can get updates about discount, availability of the product and they can buy the product.
- The graphics on the website must be catchy, to the point and must be optimized for memory and time.
- The admin must give security support so that there is no unauthorized access by earlier employees or other intruders.
- The shopping must be available on mobile app, so that all the data can be saved on cloud, and it can be accessed from anywhere.
- Finally, all the information should be stored in a database in a proper data model

Suppose you are a project manager for a software product in a new and growing market with your competitors who are also developing a product will be the same product. Which model to select and why? How will you decide the tasks and the assignment to the team members?

Case Study-4

There is a software project called “Election management system” and we want to choose a process model. What do you think is the suitable one? The software requirements are clear.

Case Study-5

Video-Rental LTD is a small video rental store. The store lends videos to customers for a fee, and purchases its videos from a local supplier. A customer wishing to borrow a video provides the empty box of the video they desire, their membership card, and payment – payment is always with the credit card used to open the customer account. The customer then returns the video to the store after watching it. If a loaned video is overdue by a day the customer’s credit card is charged, and a reminder letter is sent to them. Each day after that a further card is made, and each week a reminder letter is

Sent. This continues until either the customer returns the video, or the charges are equal to the cost of replacing the video. New customers fill out a form with their personal details and credit card details, and the counter staff give the new customer a membership card. Each new customer’s form is added to the customer file. The local video supplier sends a list of available titles to Video-Rental LTD, who decide whether to send them an order and payment. If an order

is sent, then the supplier sends the requested videos to the store. For each new video a new stock form is completed and placed in the stock file.

Case Study-6

To develop the Home Health portion of General Hospital that keeps track of the billing and Medicare requirements and payments. The current system was not scalable to the growing needs and transformation within the environment. Thus, in addition to specific desirable criteria of the chosen software, our explicit purpose in helping General was twofold: 1) to modernize their operations with current technology; and 2) to provide the best patient care available to their clients in the Home Health arena.

Case Study-7

The Income tax filing portal, will undergo changes as and when the change in the income tax rules come into effect. The requirements are clear and already existing systems may have to undergo changes.

Case Study-8

The required LMS is a digital learning environment used to support learning in schools with students from age 4 to 18. It is intended to replace an existing system (Glow) that was specially built for the purpose and which includes its own applications for e-mail, etc. Clients are from non-technical background. Users did not care about the system requirements and did not have time to interact with the development team. Secondly, stakeholders simply did not understand the terminology or approaches used – terms such as use-case simply made no sense to them.

Case Study-9

Airbus 340 flight control system is a safety critical system that implements the fly-by-wire flight system on the Airbus. The Airbus A340 is a long-haul aircraft that includes a critical fly-by-wire flight control system that mediates pilot commands and controls flight surface actuators. This is a development of the first fly-by-wire system which was introduced in the Airbus A320 aircraft.

Case Study-10

You are interacting with the MIS department of a very large oil company with multiple departments. They have a complex regency system. Migrating the data from this legacy system is not an easy task and would take a considerable time. The oil company is very particular about processes, acceptance criteria and legal contracts.

Hint: Below table will assist in making a proper selection of process model.

<u>Factors</u>	<u>Waterfall</u>	<u>V-Shaped</u>	<u>Evolutionary Prototyping</u>	<u>Spiral</u>	<u>Iterative and Incremental</u>	<u>Agile</u>
<u>Unclear User Requirement</u>	<u>Poor</u>	<u>Poor</u>	<u>Good</u>	<u>Excellent</u>	<u>Good</u>	<u>Excellent</u>
<u>Unfamiliar Technology</u>	<u>Poor</u>	<u>Poor</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Good</u>	<u>Poor</u>
<u>Complex System</u>	<u>Good</u>	<u>Good</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Good</u>	<u>Poor</u>
<u>Reliable system</u>	<u>Good</u>	<u>Good</u>	<u>Poor</u>	<u>Excellent</u>	<u>Good</u>	<u>Good</u>

<u>Short Time Schedule</u>	<u>Poor</u>	<u>Poor</u>	<u>Good</u>	<u>Poor</u>	<u>Excellent</u>	<u>Excellent</u>
<u>Strong Project Management</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Excellent</u>
<u>Cost limitation</u>	<u>Poor</u>	<u>Poor</u>	<u>Poor</u>	<u>Poor</u>	<u>Excellent</u>	<u>Excellent</u>
<u>Visibility of Stakeholders</u>	<u>Good</u>	<u>Good</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Good</u>	<u>Excellent</u>
<u>Skills limitation</u>	<u>Good</u>	<u>Good</u>	<u>Poor</u>	<u>Poor</u>	<u>Good</u>	<u>Poor</u>
<u>Documentation</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Good</u>	<u>Good</u>	<u>Excellent</u>	<u>Poor</u>
<u>Component reusability</u>	<u>Excellent</u>	<u>Excellent</u>	<u>Poor</u>	<u>Poor</u>	<u>Excellent</u>	<u>Poor</u>

Case Study 1-The most suitable model is prototyping model and evolutionary because as ABC is dealing with his first project if the project fails then we can stop at developing stages in initial stage only.

Case Study 2- I think the best model for this scenario will be the agile model this is mainly because the vague and unclear requirements.

Case Study 3-

Case Study 4- In this case, with this limited input about the case, I think we can use spiral or v-shaped Model.

Case Study 5-

Case Study 6-I think the best model for this scenario will be the Agile model.

Case Study 7-I think the best model for this scenario will be the Spiral model because of excellent knowledge in requirements.

Case Study 8- I think the best model for this scenario will be the Waterfall model.

Case Study 9- I think the best model for this scenario will be the Spiral model because of excellence in reliable system.

Case Study 10- I think the best model for this scenario will be the Incremental and iterative model because of component reusability.

Practical 4 (Requirement Development)

Objective :

To identify the various requirement development activities viz. elicitation, analysis, specification and verification for the given scenarios.

Background:

Requirement engineering produces a specification of what a system should do. The intention of requirement engineering is to provide a clear definition of requirement of the systems. This phase is a very important phase because, if the customer requirements are not clearly understood, the ambiguity can get into the other phase of the development. To avoid such issues, requirement has to be elicited using the right elicitation techniques, to be analyzed effectively, specified clearly and verified thoroughly. All activities are collectively termed as requirement development activities.

Problem Description:

Identify the requirement development activities associated with each of the following scenarios,

- a. Joe is creating an online survey questionnaire for requesting user feedback on the desired features of the application to be developed.

- b. Mark is preparing a formal document which includes all of the desired features identified by the survey.
- c. Jack identified an incomplete requirement statement.
- d. Jones is identifying all security related requirement and separating them from the performance related requirements.
- e. Merlin a team member is sent to client to observe the business case and collect typical user requirements.
- f. Leo is team member is working on requirement and ensuring that requirement collected should not be vague and unclear.
- g. Lee is conducting a facilitated meeting with the stakeholder to capture the requirements.
- h. Amit a team member is distributing questionnaires to stack holder for gathering user requirements.

Scenario	Requirement development activity
a	Elicitation
b	Specification
c	Verification
d	Specification
e	Elicitation
f	Verification

g	Elicitation
h	Elicitation