

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.

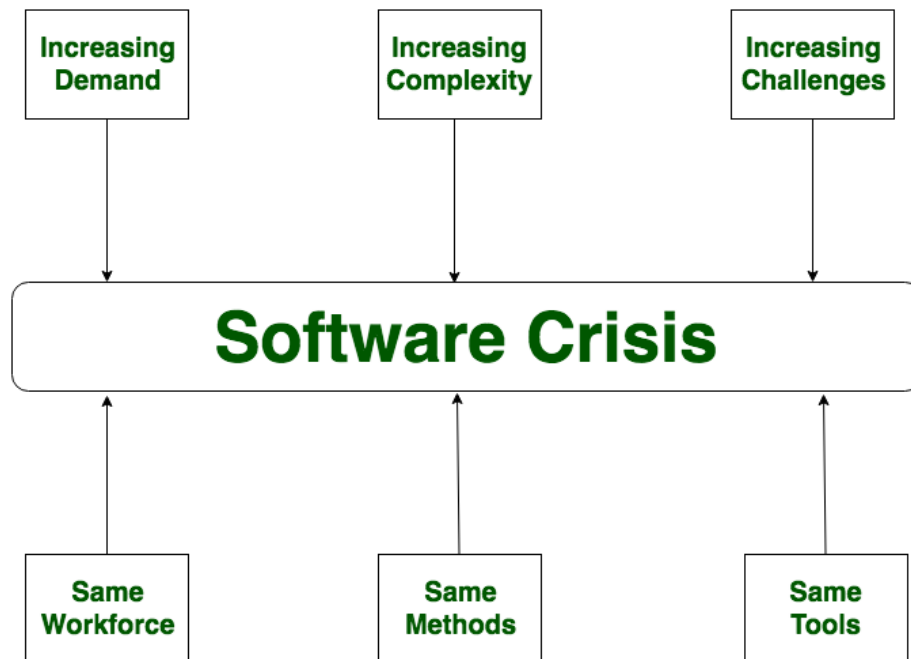
The term software crisis revolves around three concepts: complexity, change and the expectations.

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	Failure at customer Site
B	Time Slippage
C	Intractable Error after delivery
D	Cost Slippage

