**AIM: Explain the software development scenario using Extreme Programming (XP)**

**Objectives:**

·         **Brief introduction about Extreme Programming (XP)**

·         **Write one scenario**

**Introduction to Extreme Programming**

Extreme Programming (in short, XP) is an agile software development framework that was designed to improve the quality of software and satisfy the customer’s needs. It makes the life of a developer less of a hassle (no matter how constantly changes have to be made).

With extreme programming, the focus is on dynamically changing software development needs and using a well-versed development team that can apply the best technology that enables functional tests and automated units. This philosophy also believes in strong customer participation to make it possible to develop a solid software system in the shortest time.

### **Why Extreme Programming (XP)?**

Extreme Programming works towards providing iterative and recurrent software releases throughout the project; instead of everything together after a single, long project development lifecycle.

These [short iterative cycles](https://www.ntaskmanager.com/blog/what-is-agile-iterative-approach/) help both team members and customers to assess and review the project’s progress throughout its development.

### **What is Extreme Programming (XP) made of?**

#### The Values

XP incorporates the following 5 values:

* **Communication**: Software Development projects or projects in any industry rely heavily on communication. XP focuses on effective communication between the team and the customer.
* **Simplicity**: XP looks for the simplest ways to get things done. This means doing what is essential thereby reducing waste, addressing only the known issues, and keeping the design simple for effective creation and maintenance.
* **Feedback**: Feedback plays an important role in project improvement. XP encourages instantaneous feedback. This helps the team identify room for improvement and revise practices.
* **Respect**: The team must respect each other both personally and professionally, to achieve goals.
* **Courage**: XP endorses courage at all levels. This can include speaking up against what does not work and anything that affects the project’s effectiveness, or accepting feedback and improving methodologies.

## **Case Studies for Extreme Programming Practices (XP)**

### **1)    XP for Krizp System**

#### The Problem

Krizp Solution was a startup, web-based development company in India. Their business plan encompassed the creation of web portals for other small companies or educational institutions. The company began as a part-time business, employing people that were already working for other major IT organizations. The plan was to continue full-time only if the startup ventured into success. There was no framework for their software development processes as it was just a startup company with not many projects and a few employees.

The company lacked a structured approach to software development. With initial requirements jotted down on paper, further information and clarifications were received from the customer via phone calls. Usually, the major changes in requirements did not come about until the customer review, which was after the solution was developed.

Other than for [bug fixing](https://www.ntaskmanager.com/blog/defect-bug-life-cycle-in-software-testing/), the developers had little or no communication with each other. They worked separately on different features. This led to becoming a barrier to discussions regarding improvement in working methods.

Moreover, the projects were not documented. There was no project manager to track the projects or to make sure that the requirements laid out by the customer were being met. The developers worked only on what was asked to be done.

#### The Journey

The team at Krizp System was introduced to the concepts behind the different Agile frameworks. The XP method was employed over a span of one month and the results were assessed.

The CEO of the company took on 2 roles: the customer representative and the tracker. For his first role, he prioritized user stories, delegated [them to the development team](https://www.ntaskmanager.com/blog/best-scrum-tools/), and had regular communication with the customer. As the tracker, he kept track of the time to complete specific tasks. The CEO also initiated the planning game every week (or at least once in four days), as the project was small and developers could complete tasks in one user story faster. However, the customer was available for direct communication only twice per month and the rest of the time he was in contact through phone calls and e-mail.

The Paired Programming technique was adopted whereby both developers worked together. After task completion, both of the developers reviewed the code with the CEO.

Customer tests were introduced and the team worked on continuous design improvements, which were about 12-15 per month.

### **Summary**

The XP approach seemed to have a good impact on the software development cycle for the company. Some of the positive changes included:

1. Better team collaboration, communication, and feedback
2. Better task and time management, and
3. Increased CEO involvement without technical contribution.

## **2) Extreme Programming Practices for IBM and Sabre Airlines**

#### The Problem

To assess the practical [applications of Waterfall](https://www.ntaskmanager.com/blog/agile-vs-waterfall/) vs. Extreme Programming, a research study was conducted through 2 case studies: one at IBM and the other at Sabre Airlines. Each case study compared the waterfall approach to the XP approach.

#### The Journey

In the first case study, at IBM, the researchers wanted to study the impact of adopting the XP approach on productivity, quality, and customer satisfaction. A year-long study was conducted on a team of 7 – 11 members regarding the adoption of XP practices. The team was responsible for developing Servlet/XML applications for a toolkit utilized by other IBM teams to create products for external customers. The case study analyzed 2 approaches on consecutive releases of the same product. The first one was the traditional waterfall approach and the second was XP.

In the second case study, at Sabre Airline Solutions, the same method was used i.e. comparing 2 approaches through different releases of the same product. The team worked on developing a scriptable GUI environment for external customers to develop customized end-user and business applications. The team comprised 6-10 members. The old release was finished 3 years prior (spanning 18 months) using the waterfall method whereas the new release was completed recently (spanning 3.5 months), using XP.

The first step was to establish an Extreme Programming Evaluation Framework (XP-EF), which comprised three parts: XP Context Factors (XP-cf), XP Adherence Metrics (XP-am), and XP Outcome Measures (XP-om):

* **XP Context Factors (XP-cf)**: XP-cf was used to record important information related to the project. These factors included team size, project size, criticality, and staff experience.
* **XP Adherence Metrics (XP-am)**: Through XP-am, the extent to which the team uses the XP practices was expressed. The XP-am also helped in investigating the interactions and dependencies amid the XP practices as well as the degree to which the practices can be detached or removed.
* **XP Outcome Measures (XP-om)**: XP-cm enabled assessment of business-related results i.e. productivity, quality, etc.

In addition to the framework, interviews were conducted with team members and customers to help understand the incorporation of XP by the team for the customer’s satisfaction.

### **Summary**

At IBM, the XP method seemed more productive compared to the waterfall method by the following measures:

* **Test Defects**: for pre-release, the defects were 50% lesser and for post-release, the defects were about 40% lesser in the release through the XP approach.
* **Productivity**: There was a significant increase in staff productivity using the XP approach than in the waterfall method.
* **Customer Satisfaction**: Customer satisfaction was noted to be high in XP and documented as N/A for the waterfall.
* **Morale**: The morale of the stakeholders was recorded as high in XP and documented as N/A for the waterfall.

At Sabre Airlines, similar results were noticed:

* **Defect Collection Period**: As the first release was created over 18 months, the defect collection period was also longer in the waterfall-based approach. It was significantly shorter in XP base release.
* **Test Defects**: for pre-release, the defects were 65% lesser and for post-release, the defects were about 46% lesser in the release through the XP approach.
* **Productivity**: Staff productivity using the XP approach was about 46% higher than in the waterfall method.
* **Customer Satisfaction**: Customer satisfaction was noted to be high in XP and documented as N/A for the waterfall.