Continuous Comprehensive Assessment Report

On

"Extreme Programming and Adaptive Software Development"

A CCA report for the course Software Engineering and Project Management (21CS61)

submitted in partial fulfillment of the requirements for the Continuous Comprehensive

Assessment (CCA)

Submitted by

Team Members: Swara   (1dt21ai058)
Nikitha (1dt21aio40)
Soumya (1dt21ai054)

Course Coordinator:
Mrs. Mahalakshmi G
Asst. Prof
Dept of AIML

DAYANANDA SAGAR ACADEMY OF TECHNOLOGY & MANAGEMENT
Opp. Art of Living, Udayapura, Kanakapura Road, Bangalore- 560082
Affiliated to Visvesvaraya Technological University,
Belagavi and Approved by AICTE,
New Delhi

CONTENTS

CHAPTER 1

INTRODUCTION

PRINCIPLES OF EXTREME PROGRAMMING

Constant and early feedback: Feedback is a traditional approach followed in the waterfall software development methodology. XP embraces change, and the teams strive to receive early and constant feedback.

Assumed simplicity: Developers need to focus on tasks with the highest priority and maximum business impact.

Incremental changes: It is about making small and iterative changes to the product rather than building the entire product in one go.

Embracing change: It is all about considering the customer's viewpoints and opinions and then adapting to the change.

Quality work: The team collaborates closely to deliver quality output within stipulated timelines.

ADAPTIVE SOFTWARE DEVELOPMENT

In ASD, as in most agile methodologies, work occurs in cycles or iterations. During such cycles, developers not only build new components but also make necessary modifications to pre-existing ones. A key distinction between adaptive software development and other methodologies is that, in ASD, cycles are component-based, rather than task-based. It is common, for instance, for teams using other methodologies to break down into more granular tasks, which are then assigned to developers to implement. In ASD, the focus is always the desired result. It defines components as a group of features, planned, implemented and delivered together.

CHAPTER 2

CASE STUDY

ADS-CASE STUDY

Implementing ASD in a Healthcare Management System

Project Background: A healthcare provider needed a new system for patient data, appointments, billing, and communications, requiring flexibility to adapt to changing regulations.

Phase 1: Speculation

Gathered initial requirements through stakeholder workshops.

Created a high-level project plan.

Formed cross-functional teams.

phase 2: Collaboration

Used short iterations (2-4 weeks) for development.

Held regular feedback sessions with stakeholders.

Conducted pair programming and peer reviews.

Created prototypes for early validation.

Phase 3: Learning

- Conducted retrospectives after each iteration to reflect and improve.
- Documented and shared lessons learned.
- Continuously refined project plans based on feedback.

CHAPTER 3

CONCLUSIONS

Extreme Programming (XP) is a method of improving software quality and response to changing needs through continuous delivery, collaboration, and practices like test-driven development. Its goal is to allow small to mid-sized teams to produce high-quality software and manage changing customer requirements.

What sets it apart from the other agile methodologies is that XP focuses on the technical aspects related to software development. It is all about following the best practices, as the right engineering practices can go a long way in delivering high-quality code at a sustainable pace.

It introduced a newer method of innovation and growth in the software development ecosystem.

It was in the year 1997 when Ken Beck, a software engineer and an active contributor to the agile manifesto, created extreme programming. Organizations have relied on XP concepts to accomplish their software development goals.

REFERENCES

[1]https://www.geeksforgeeks.org/adaptive-software-   development-asd/

 [2]https://www.researchgate.net/publication/3895404_Case_sudy_Extreme_programming_in_a_univert

[3]https://link.springer.com/book/10.1007/b137278

[4]https://www.ijert.org/research/case-study-of-agile-methodologies-in-field-of-software-development-IJERTV6IS110104.pdf