


DIPLOMA WALLAH

(Your One Stop Hub For Diploma Resources)

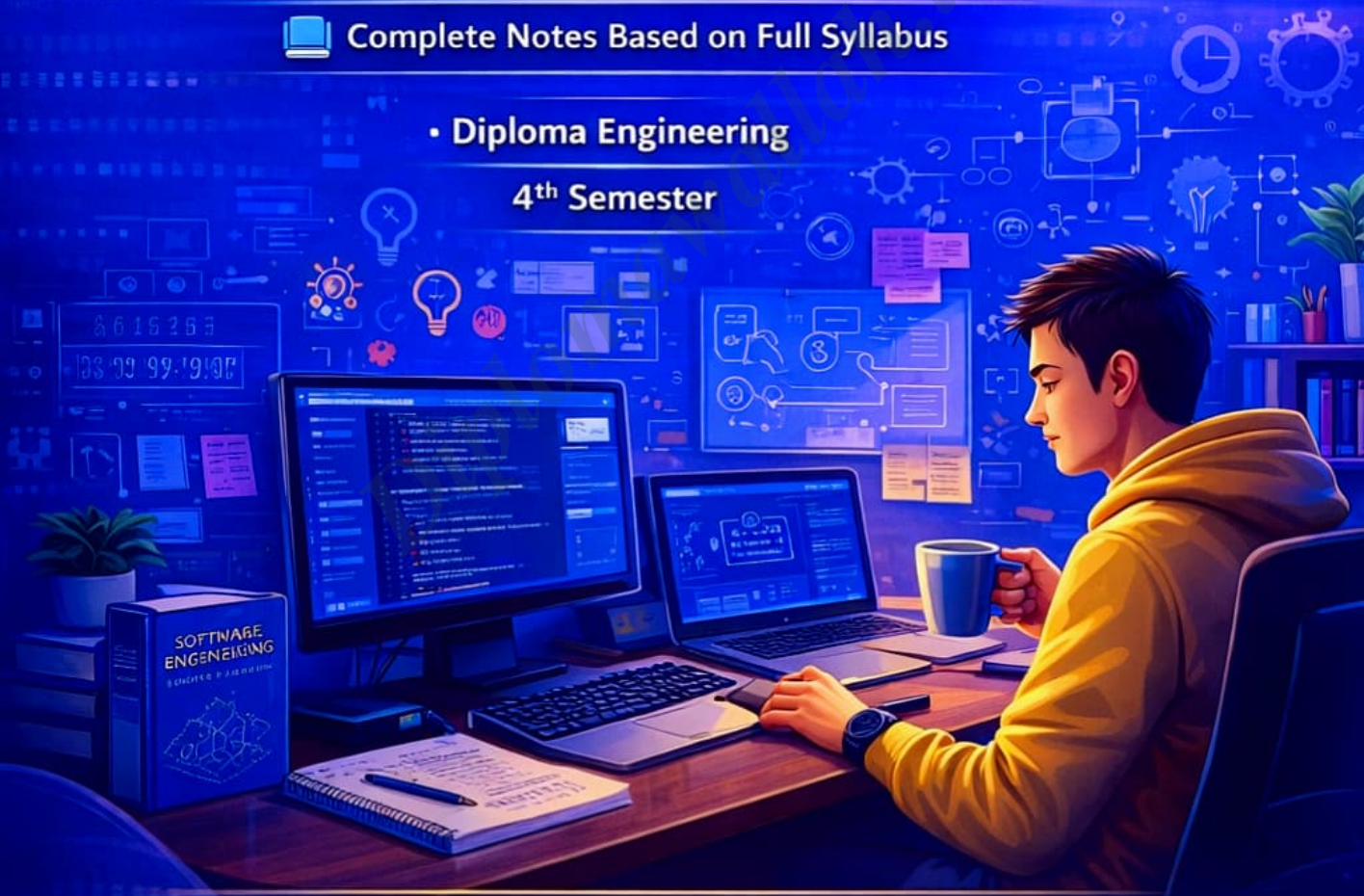


SOFTWARE ENGINEERING

PRINCIPLES AND PRACTICES

 Complete Notes Based on Full Syllabus

• Diploma Engineering
4th Semester



© Diploma Wallah. All Rights Reserved

Unauthorized sharing/selling is strictly prohibited

www.diplomawallah.in

Notes prepared by Sangam

unit-03

Agile frameworks:-

Agile frameworks are structured methodologies designed to implement the Agile principles of flexibility, collaboration, and continuous improvement in project management.

These frameworks include:-

i) Scrum

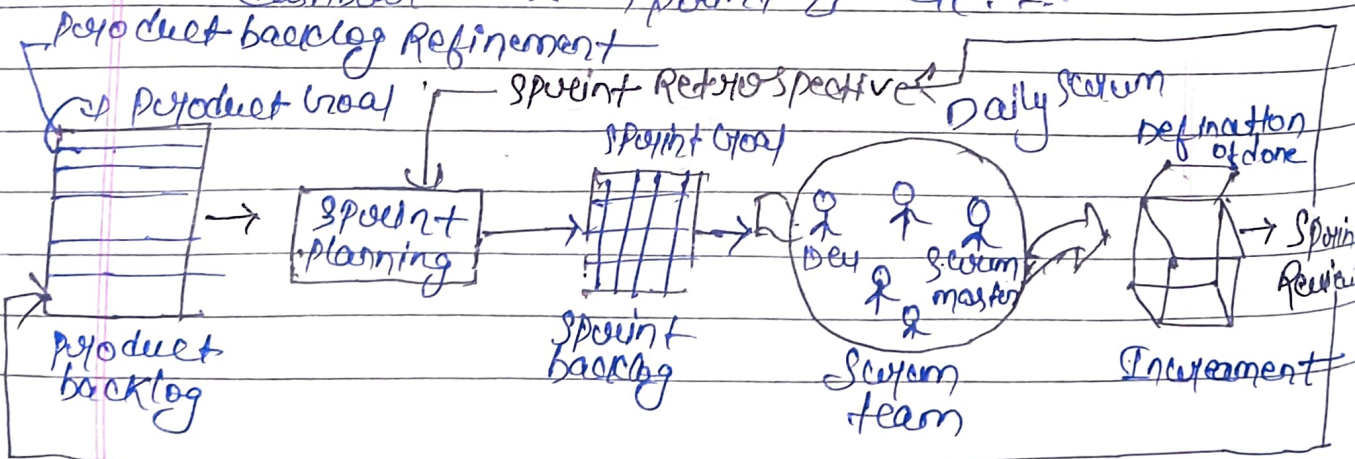
A popular framework that is used whenever we want to develop complex products.

- work is divided into sprints (usually 2-4 weeks)
- After every sprint, ^{we} get a working product.

Key role

- Product owner - Tells what needs to be built.
- Scrum master - makes sure team follows Agile and remove problems.
- Development Team - Build the product (Developer, tester, designer all are working).

Example:- Imagine we are building a mobile app. In each sprint, our team will complete one major feature - like login in sprint 1, dashboard in sprint 2 etc...



ii) Kanban:-

- It is a popular Agile software development methodology (work in progress limit).
- Focuses on visualising work using a Kanban board (To do → in progress ^{Testing} → done)
- No fixed sprints; tasks are added and moved continuously.
- Helps see bottlenecks and improve flow.

Example:-

In a support team, tickets are constantly coming. They just keep moving tasks across the board as they are completed. There is no sprint.

* When needed

→ dynamic/frequent changing req. which needed to deliver fast.

→ frequent release are there (periodically).

→ when incoming work is continuous.

* Kanban Board/board

It is critical to understand the visualisation of workflow stages in the task execution pipeline. Kanban board provides a simple way to understand the process. It can be explained

* Every request received is put on the Kanban board.

* A column on the board represents a stage (working stage).

* The received stage could be called "backlog" also.

* The team could decide the names for the phases based on the terminology used by respective teams.

* ALM tools like Rally/Jira could be configured to use.

Simple Kanban board

Work stage	Backlog	Acknowledged	Development	Testing	Deployment	Done	
Status	-	-	In Progress	Ready	In Progress	Ready	-
Jobs	☐ ☐	☐ ☐	☐ ☐	☐ ☐	☐ ☐	☐ ☐	☐ ☐

column ↑

lane across column.

work item ↓

- ☐ work item (Type 1)
- ☐ work item (Type 2)

↓ Simplicity, communication, feedback, respect.

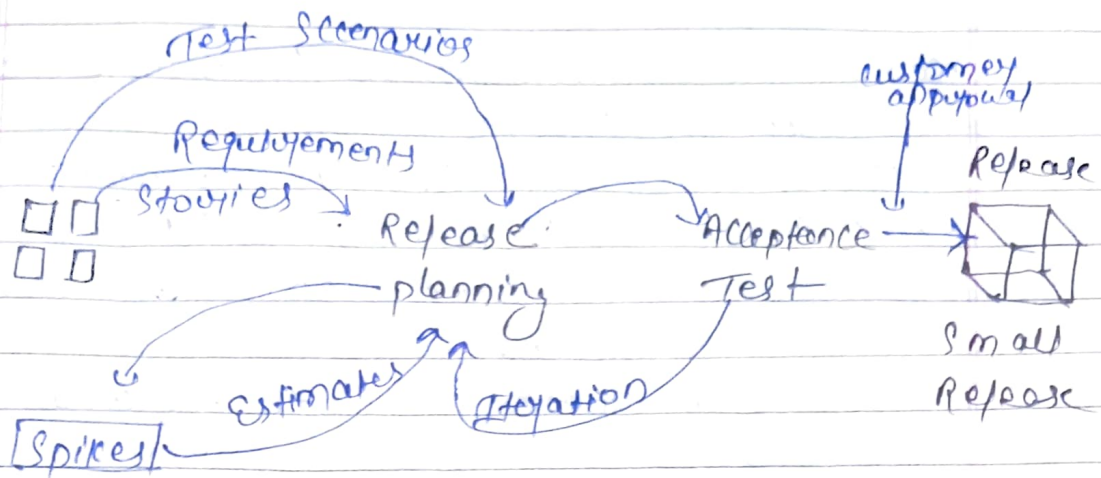
- iii) Extreme Programming (XP)
- * popular software dev. framework of Agile models.
- * used to improve SW quality and responsiveness to customer req.
- * It focus on delivering high-quality SW through frequent and continuous feedback, collaboration, and adaption.

focuses on technical practice like:-

- Test-driven development (TDD)
- pair programming, continuous integration.

Ex - In banking app, code must be 100% accurate.
XP helps ensure the code is always tested and safe before release.

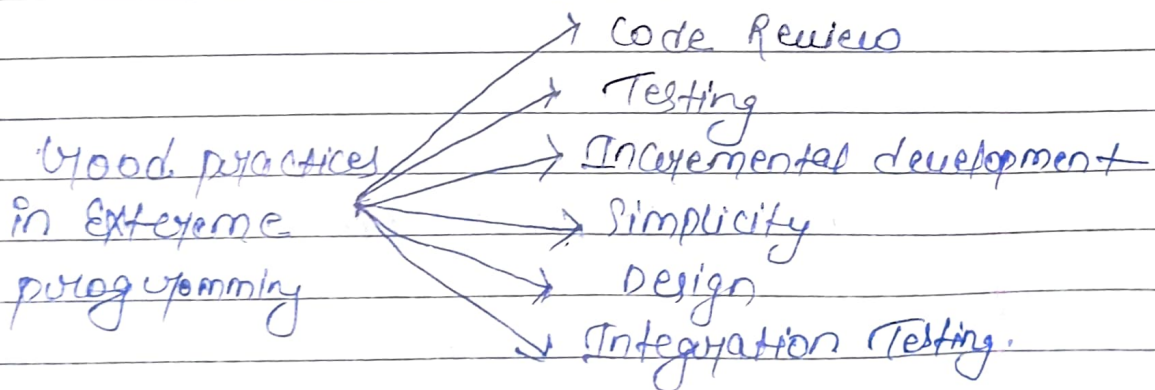
philestikho
for code
code developer
it etc
computer
artisan
cash chain



Some Common principle:-

1. Working S/W is the key measure of progress in project.
2. For progress in a project, therefore S/W should be developed and delivered rapidly in small increments.
3. Even late changes in the req. should be accepted.
4. Face to face communication preferred.
5. Continuous feedback involved.

* Good practices in Extreme programming.



- **Code Review**:- Code reviews detects and corrects errors efficiently.
- **Testing**:- Testing code help to remove errors and improves its reliability. XP suggest test-driven development (TDD) to continually write and execute test cases.
In TDD approach test cases are written even before any code is written.
- **Incremental development**:- Customer feedback gained and based on this development team comes up with new increments every few days after each iteration.
- **Simplicity**:- Simplicity makes easier to develop good quality code.
- **Design**:- Good quality design is important to develop good quality software. So everybody should design daily.
- **Integration Testing**:- Integration Testing helps to identify bugs at the interfaces of different functionalities.

planning → design → coding → testing → listening

life-cycle -

Sangam