TABLE OF CONTENTS

USER STORIES FOR REQUIREMENTS

- Sitting down with customer.
- Make list of features they would like to see.
- It is our to-do list for the project.
- List is prioritized according to the customer.
- Most important first, least important last.

REFACTORING

- It the process of restructuring of existing code, to a much easier and understandable code.
- Programming team look for possible improvements and make those improvements even when there is no immediate need for them.
- Later changes are easier to make because the code is well-structured and clear.

TEST – FIRST DEVELOPMENT

Test-first Development

- Testing is central to XP and XP has developed an approach where the program is tested after every change has been made.
- XP testing features:
  - Test-first development.
  - Incremental test development from scenarios.
  - User involvement in test development and validation.
  - Automated test harnesses are used to run all component tests each time that a new release is built.

Test-driven Development

Writing tests before code clarifies the requirements to be implemented.

Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.

- Usually relies on a testing framework such as Junit.

All previous and new tests are run automatically when new functionality is added, thus checking that the new functionality has not introduced errors.

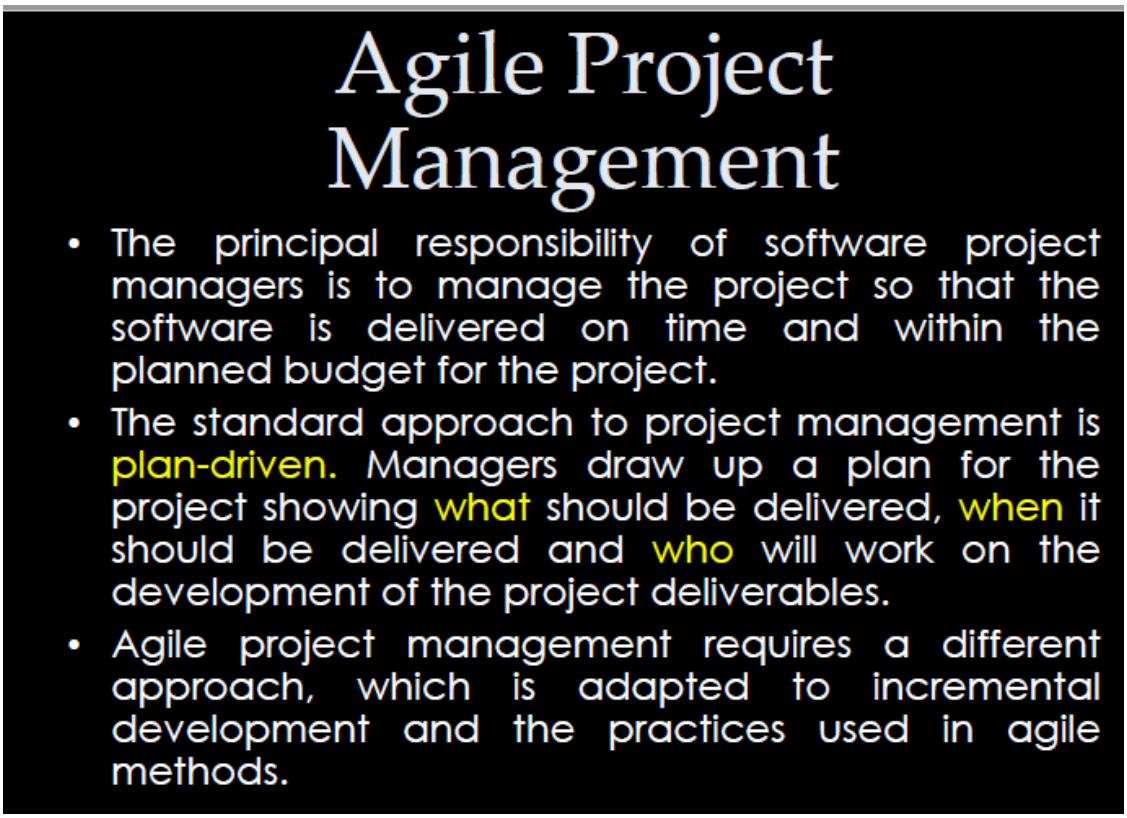
PAIR PROGRAMMING

- In pair programming, programmer sit together at the same computer to develop the software.
- Pairs are created dynamically so that all team members work with each other during the development process.
- Some evidence suggests that pair a working together is more effective that 2 programmers working separately.

PAIR PROGRAMMING ADVANTAGES

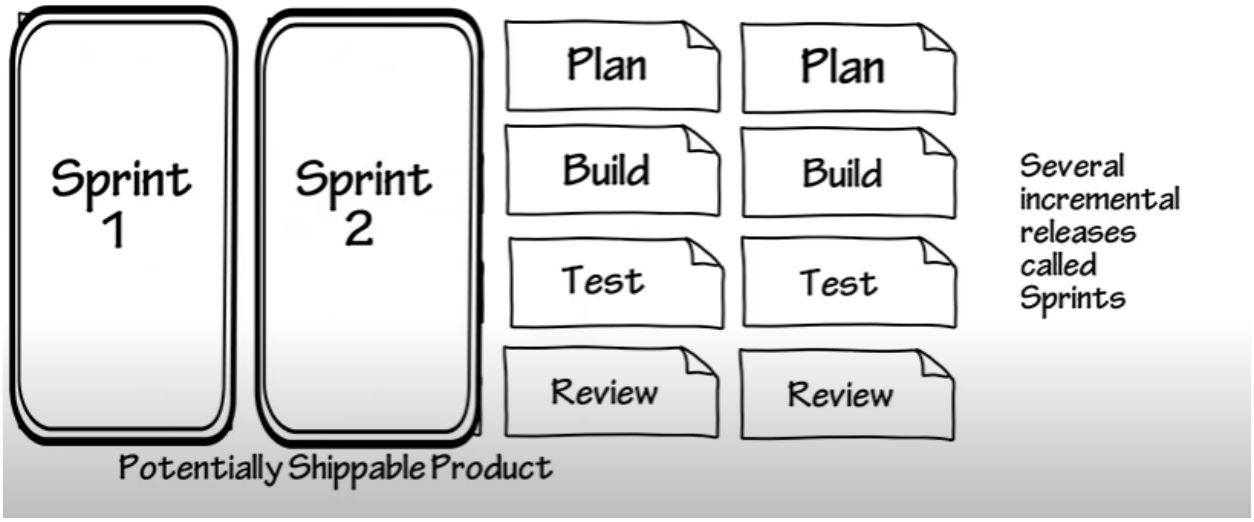
- The sharing of knowledge that happens during pair programming is especially important as it reduces the overall risks to a project when team members leave.
- This helps develop common ownership of code and spreads knowledge across team.
- It serves as informal review process as each line of code is reviewed by more than one person.
- It encourages refactoring as the whole team can benefit from improving the code.

AGILE PROJECT MANAGEMENT



SCRUM

- Product is released in several increments called Sprints.



- Sprints keep repeating until product is completed.
- It is a team-centric methodology and there is no certain leader.
- Everyone is responsible for everything.

3 ROLES

1. **Product Owner**
   - Responsible for defining the features that are needed in the product.
2. **Scrum Master**
   - Responsible to overlook the process and protect the team, and keeping things going.
3. **Team**
   - Testers, Developers, Writer, and anyone needed in developing the product.
   - Anyone can do each other's work if capable.

3 ARTIFACTS

1. **Product Backlog**
   - It consists of User-Stories arranged in accordance to priority.
2. **Sprint Backlog**
   - Highest prioritized user-story goes into the sprint backlog and is committed to the next sprint.
3. **Burndown Chary**
   - It shows progress during a sprint.
   - It should reach 0 on completion of a sprint.

3 CEREMONIES

1. **Sprint Planning**
   - Planning before starting a sprint.
2. **Daily Scrum**
   - Daily checkup to see if there are any problem in the current sprint.

3. **Sprint Review**
    - Last review on the complete scrum and look for any improvement for the next sprint.