Q1: Read about “Planning Poker” - Agile estimation technique and illustrate an example with a Development Team of 10 who are tasked to develop a mobile app for Maha-Khumb in 3 months.

Planning Poker is an estimation technique used in Agile development to estimate the effort required to complete tasks in a project. It is a consensus-based method where team members assign story points to user stories based on complexity, uncertainty, and effort.

Example Scenario: Development Team of 10 for Maha-Khumb Mobile App

A team of 10 developers has been assigned to develop a mobile app for Maha-Khumb within 3 months. The team follows Scrum methodology and uses Planning Poker to estimate user stories.

Step-by-Step Execution of Planning Poker

1. User Stories are Defined:
   The Product Owner presents user stories, such as:
   - Story 1: User registration and authentication
   - Story 2: Interactive map of Maha-Khumb event locations
   - Story 3: Live streaming of important Maha-Khumb events
   - Story 4: Push notifications for event updates

2. Team Members Assign Story Points:
   Each team member gets a deck of cards with Fibonacci sequence values (1, 2, 3, 5, 8, 13, 21, etc.). Each member independently selects a card representing their estimate of effort and complexity.

3. Discussion and Consensus:
   - Suppose for Story 2 (Interactive Map):
     - Some developers assign 5 points (moderate effort).
     - Others assign 8 points (complexity in GPS integration).
   - The team discusses reasons for the differences.
   - After discussion, they agree on 8 points for this story.

4. Final Estimations:
   The process is repeated for all user stories.
   - Story 1 (User Registration) → 3 points
   - Story 2 (Interactive Map) → 8 points
   - Story 3 (Live Streaming) → 13 points
   - Story 4 (Push Notifications) → 5 points

5. Sprint Planning:
   Based on total available effort, the team plans sprints to complete the most critical features first within the 3-month timeline.

Conclusion

Planning Poker helps the team make collaborative, realistic, and unbiased estimates while ensuring everyone understands the work involved. This technique improves transparency, engagement, and accuracy in Agile projects.

Q2: Read Paper – Measuring Software Development Waste in OSS Projects -https://arxiv.org/pdf/2409.19107. Pick one measure from this paper and apply it on any open-source repository. Share results.

After reading the above paper, I am choosing PR Rejection Rate (PRR) and applying it to a GitHub Project.

The PRR is calculated as:

Unmerged PRs/Total Closed PRs.

The repository that I am choosing to apply the above measure is:

https://github.com/Trinea/android-open-project/pulls

In this repository:

Unmerged PRs= 17

Closed PRs= 291

PRR= 17/291

PRR= 0.0584