What is **planning poker?**

- Planning Poker, also called “Scrum Poker,” is a consensus-based Agile planning and estimating technique used to assess product backlogs, guessing how much time and effort is needed to complete each of the backlog’s initiatives.
- It’s called “Poker” because everyone uses physical cards that resemble playing cards.
- The cards estimate the number of story points for each task or backlog story being discussed.

Cards in planning poker

- This Poker tool cards are assigned numerical values loosely based on the Fibonacci sequence
- However, sometimes the Poker tool uses this sequence: 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100.
- Some Planning Poker decks also include three additional cards, showing an infinity symbol, a question mark, and a coffee cup.
- The infinity symbol (∞) represents “This item is too big”.
- Question mark(?) to show that they don’t understand the item and wish to ask the product owner additional questions.
- Coffee cup: I’m tired and hungry and want a break.

How to **play**

- **Product Owner Presents the User Story** : The team discusses the feature/task.
- **Team Asks Questions** : Developers clarify doubts to understand scope.
- **Each Member Chooses a Card** : Team members reveal their estimates at the same time.
- **Discussion & Consensus** : If estimates vary widely, members discuss and justify their reasoning.
- **Repeat Until Agreement** : The process continues until the team reaches a consensus

Maha-Kumbh Mobile App Development

- **The Product Owner provides a backlog of features**:
- User Registration
- Event Schedule & Notifications
- Route Maps & Navigation
- Emergency Contacts
- Vendor/Accommodation Listings
- Push Notifications & Alerts

Route map and Navigation

Developer	Estimate (Story Points)
Dev 1	8
Dev 2	5
Dev 3	13
Dev 4	8
Dev 5	5
Dev 6	8
Dev 7	8
Dev 8	5
Dev 9	8
Dev 10	13

- Some developers estimated **5**, while others estimated **13**.
- Those who selected **5** believed existing libraries would reduce effort.
- Those who chose **13** pointed out challenges like **network bandwidth**, **server load**, and **scalability issues**.
- After discussion, the team **agrees on 8 story points**.

Measuring Software Development Waste in Open-Source Software Projects

Metrics

- Stale Fork
- Project Diversification Index
- Pull Request (PR) Rejection Rate
- Backlog Inversion Index
- Feature Fulfilment Rate

Pull Request (PR) Rejection Rate.

- Pull Requests (PRs) facilitate collaboration by allowing team members to provide feedback and approve modifications, ensuring that new additions meet the project's standards and requirements.
- An accepted PR typically has undergone successfully passed code reviews, where other developers verify that the changes are error-free, align with the project's coding standards, and contribute positively to the functionality and objectives of the application.
- The PR Rejection Rate indicates the proportion of work that does not make it into the project compared to the work that did, indicating 'unused artifacts'.

Application to an Open-Source Repository: NUMPY

- PR rejection rate=(Number of Rejected PRs/Total Number of PRs)×100%

Total number of PRs=10000

Rejected PRs =1500

PR rejection rate= (1500/10000)X100=15%