

DEVOPS ASSIGNMENT

“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.”

**Sarthak Rawat (205224021)
NITT**

Planning Poker

Planning poker, also known as Scrum poker, aids agile teams in estimating the time and effort required to complete each initiative on their product backlog. The term "planning poker" originates from the gamified technique where participants utilize physical cards resembling playing cards to estimate the number of story points for each backlog story or task under discussion. The purpose of this process is to assist software organizations in more accurately estimating development timeframes, fostering consensus among cross-functional team members, and strategically planning the team's work.

Functioning:

Planning poker brings together stakeholders from various departments within the organization to reach a consensus on the estimated effort needed for multiple backlog initiatives. In an agile software organization, stakeholders may include a product owner, developers, UX designers, QA testers, and product managers, among others.

Steps:

Distribute the cards

All participants receive an identical deck of cards (or chips), each bearing a different number. One common sequence involves doubling each number: 1, 2, 4, 8, 16, 32, 64. Another sequence, recommended by Mike Cohn of Mountain Goat Software who popularized planning poker for agile development, consists of 0, 1, 2, 3, 5, 8, 13, 20, 40, and 100. The decks are intentionally limited, featuring significant number-jumps, to facilitate all participants in reaching a consensus number for each story. Providing participants with too many options, such as every number from 1 to 50, would render the process inefficient.

Review the story

Subsequently, the product owner (or a product manager) will orally present each story to the group.

Deliberate

Upon hearing the story, the group engages in a discussion. Participants articulate how they envision tackling the work, estimate the number of

individuals required, identify necessary skill sets, anticipate potential obstacles impeding progress, and pose any questions regarding the story.

Estimate and disclose

Once all participants have shared their perspectives and queries have been addressed, each individual privately selects a card from the deck to represent their estimate of story points. Upon readiness, all participants reveal their cards simultaneously. A higher card value indicates the participant perceives the story as more challenging to complete.

Strive for consensus

If all participants unveil the same card, that number becomes the consensus, allowing the group to proceed to the next story. In cases where cards differ, the group continues discussing the story. Individuals with higher (or lower) estimates than the rest of the group elucidate their rationale and endeavor to persuade their colleagues to consider their standpoint. Following this additional round of discussion, all participants review their cards and either maintain their initial choice or opt for a new one. Subsequently, all participants reveal their cards simultaneously once again.

Story:

User downloads trial version of app

Estimators	Round 1	Round 2
Matt	5	8
Kirsten	5	8
Ivan	8	8
Sandra	13	8

Example: Estimating Work for a Maha-Khumb Mobile App Using Planning Poker

Maha-Khumb Mobile App:

The Maha-Khumb Mobile App is designed to assist millions of pilgrims and visitors attending the Maha-Khumb event. The app will furnish real-time information, navigation support, and emergency aid to ensure a seamless and organized experience for users.

Key Features of the App:

This application will encompass:

Event Schedules – A comprehensive timetable of religious ceremonies, cultural events, and significant gatherings.

Live Updates & Notifications – Immediate alerts for schedule modifications, weather updates, and emergency declarations.

Navigation & Maps – GPS-enabled maps guiding users to various locations including temples, ghats, medical facilities, and food stalls.

Emergency Contact Information – Swift access to police, medical assistance, and lost-and-found services.

Multi-language Support – Availability in multiple languages to accommodate visitors from diverse regions and countries.

Development Team and Project Timeline

A team of 10 developers has been tasked with developing this app within a 3-month timeframe. To ensure efficient and well-planned development, the team employs Planning Poker for effort estimation.

Breakdown of Work

The development team, alongside the Product Owner, divides the work into smaller tasks or user stories:

"As a user, I wish to view event schedules to plan my visit."

"As a user, I wish to receive live updates to stay informed."

"As a user, I wish to access an interactive navigation map to locate places."

"As a user, I wish to call emergency services when necessary."

"As a user, I wish for the app to support multiple languages for ease of use."

Conducting Planning Poker for Effort Estimation

The team selects the initial user story:

"As a user, I wish to view event schedules to plan my visit."

The team deliberates on the complexity of fetching event data, UI design, and real-time updates implementation.

Each team member selects a card (without revealing it to others).

Upon cue, all members simultaneously disclose their chosen numbers.

First Round – Initial Estimates

3 members opt for 5 points (deeming it a simple UI feature).

4 members select 8 points (considering API integration complexity).

3 members choose 13 points (expressing concerns about real-time updates and time zone management).

Discussion among members

The members who chose 13 elucidate that real-time data synchronization and time zone disparities could heighten implementation complexity.

The members who selected 5 argue that storing event data locally could reduce complexity.

Following further deliberation until a middle ground is reached.

Re-estimation and Consensus

The team revotes, with the majority now opting for 8 points as a fair estimate.

This number is finalized for the task.

Repeating the Process for Other Features

The team repeats Planning Poker for all remaining user stories:

Task: "As a user, I wish to access an interactive navigation map to locate places."

The team discusses GPS, map APIs, and real-time location tracking.

Upon estimation, they determine this task necessitates 13 points.

Task: "As a user, I wish for the app to support multiple languages."

Given the translation of UI elements and text, they settle on 8 points.

By adhering to this estimation process, the team ensures that development tasks are meticulously planned, realistic, and achievable within the 3-month timeline.

DEVOPS ASSIGNMENT

**“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.”**

Sarthak Rawat (205224021)

NITT

Metric: Feature Fulfillment Rate (FFR)

Feature Fulfillment Rate (FFR) measures how many requested features (via issues) are successfully implemented over time. It helps gauge how efficiently a project responds to user needs.

OSS Project

<https://github.com/harmonoid/harmonoid>

Applying FFR to Harmonoid

Data Collection:

From the [harmonoid/harmonoid](https://github.com/harmonoid/harmonoid) GitHub repository (as of March 5, 2025):

- Total Feature Requests (Labeled "Enhancement"): 28
- Implemented Features (Closed with Fix): 16
- Pending Feature Requests: 12

Calculation:

FFR can be calculated using the formula: $FFR = (\text{Implemented Features} / \text{Total Feature Requests}) \times 100$. In this case, the FFR is determined to be 57.14%.

Interpretation:

- The Feature Fulfillment Rate (FFR) is ~57%, meaning a little over half of requested features have been implemented.
- This suggests that while the project is actively addressing user requests, some backlog remains.