MTech Through-Train
Certificate in Intelligent Reasoning Systems (IRS)
Practice Module 2025

Group 7: SprintPredict

Full Name	Student ID	Email
Harshit Agarwal	A0315374E	E1509813@u.nus.edu
Chitrarath Bhattacharjee	A0315514M	E1511452@u.nus.edu
Gan Jia Hui	A0260540R	E1092587@u.nus.edu

Problem Statement: AI-Driven Sprint Planning and Prioritization System

Background & Challenge

Agile teams leverage tools like JIRA, Monday.com, ClickUp etc to manage product backlogs and plan sprints. However, sprint planning remains a largely manual, time-intensive, and subjective process, resulting in several inefficiencies:

- **Inefficient Workload Distribution:** Sprint allocation is often performed manually, leaving significant room for optimization. Achieving an optimal allocation plan that considers team members' availability, capacity, skill sets, preferences, task dependencies, and velocity is complex and challenging.

- **Subjective Sprint Planning:** Decisions during sprint planning frequently rely on intuition rather than empirical data or business impact analysis, leading to inconsistent workload assignments.

- **Lack of Data-Driven Insights:** The absence of predictive analytics and insights results in teams either overcommitting or undercommitting, often overlooking individual preferences and workload balance.

Proposed Solution

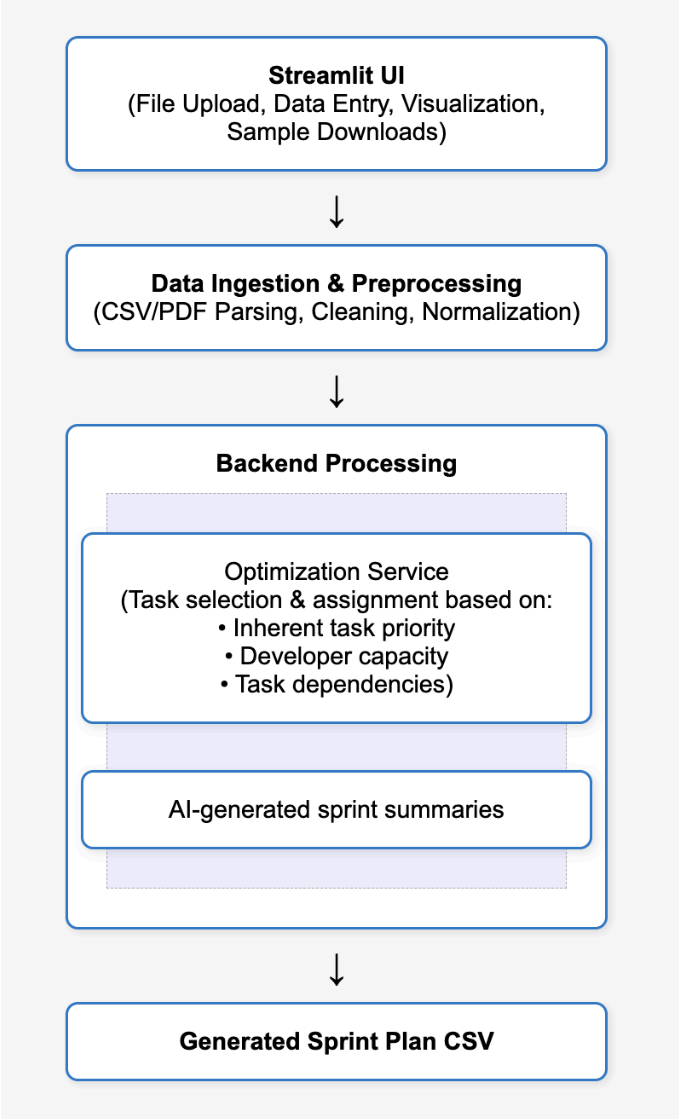
We propose an **AI-powered Sprint Planning and Prioritization System** that will:

✅ **Analyze past sprint data** – Examine data from past sprints to determine trends, team velocity, and factors of slippage (team members overestimating, tickets moving from one sprint to another and individual developer overcommitment). Use this data to forecast potential slippage in upcoming sprints.

✅ **Factor in team constraints** – Consider **team strength, availability, skill sets, dependencies, and past performance** to recommend an optimal sprint backlog with explainability.

✅ **Generate optimal sprints** – Automatically suggest sprint compositions, ensuring a balanced workload distribution across team members, along with AI-generated sprint summaries using NLP.

Architecture



Streamlit UI
(File Upload, Data Entry, Visualization, Sample Downloads)
Data Ingestion & Preprocessing
(CSV/PDF Parsing, Cleaning, Normalization)
Backend Processing
Optimization Service
(Task selection & assignment based on:
• Inherent task priority
• Developer capacity
• Task dependencies)
AI-generated sprint summaries
Generated Sprint Plan CSV

1. **User Interface (Streamlit UI):**
   - Provides an interactive web interface for file uploads (CSV), manual data entry
   - Displays data, including raw data, processed data, and the final sprint plan.

2. **Data Ingestion & Preprocessing**
   - Reads and cleans data from product backlogs, historical sprint records, and team details.
   - Normalizes and structures the data to prepare it for downstream processing.

3. **Optimization Service:**
   - Formulates a Mixed-Integer Linear Programming model to select and assign tasks.
   - Considers inherent task priorities, developer capacity, dependencies, and skill matching.
   - Generates an optimal sprint plan.

4. **Explainability & Reporting:**
   - Produces AI-generated sprint summaries
   - Provides transparency into task selection, workload distribution, and assignment decisions.
5. **Output:**
   - The final sprint plan is provided as a downloadable CSV file, detailing task assignments and sprint compositions.

Approach for an AI-Driven Sprint Planning and Prioritization System

1. Data Ingestion & Preprocessing

- **Data Sources:**
  - **Product Backlog:** Import a comprehensive backlog (200+ user stories/tasks) with details like Issue Key, Summary, Description, Priority, Story Points, and Dependencies.
  - **Historical Sprint Data:** Ingest data from past sprints (e.g., from Jira, Monday.com) that include details on team velocity, task completion status, and overall sprint data(story point committed, story points completed)
  - **Sprint Team Data:** Load details on team members, including their availability, capacity (in story points for each future sprint), skill sets, and preferences
  - **Preparation:**
    - Clean and normalize data.
    - Parse dependencies, convert priorities into standardized metrics, and ensure consistency across all data sets.
    - Assign AI analysed skill set to each task, which needs to be divided into sprints.

2. Analysis of Historical Sprint Data

- **Trend Analysis:**
  - Examine past sprint data to extract trends such as average team velocity, developer-wise average overcommitted story points, sprint sprint-wise average spillage(story points)
- **Forecasting:**
  - Use statistical analysis and machine learning techniques to forecast potential slippage in upcoming sprints based on historical performance.
- **Capacity Calibration:**
  - Calculate the effective capacity of each team member and determine a safe planning threshold (e.g., 70% of capacity) to leave room for ad-hoc work.

3. Factoring in Team Constraints

- **Team Strength & Availability:**
  - Consider current availability, workload, and developer preferences to ensure fair distribution of tasks.
- **Skill Set Matching:**
  - Analyze the required skills from user stories and match them against the developer skill sets to ensure that tasks are assigned to the most qualified team members.
- **Dependency and Task Complexity:**
  - Incorporate dependencies between tasks, ensuring that prerequisites are completed or scheduled appropriately.

4. Optimization Model for Sprint Planning

- **Devising and optimisation model for sprint planning**

5. NLP-Driven Sprint Summarization and Explainability

- **Explainability:**
  - Leverage NLP techniques to produce human-readable sprint summaries that explain why specific tasks were chosen, how the workload was balanced, and how dependencies and skill requirements were addressed.