SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE PROCESS MODEL DOCUMENT

PROJECT NAME: LIKE

GROUP MEMBERS: Eylül ÖZTÜRK, Ahmet Melih MOR, Burak GÜLERYÜZ, Ali Berat AKOĞLU, İmge Sümbül YÜKSEL, Kayra Mert AKDAĞ

#	NECESSARY NEEDS FROM THE ORGANIZATIONAL PROCESS
1	**Clear Objectives:** A well-defined project scope and objectives are essential for guiding the project team and stakeholders towards a common goal.
2	**Effective Communication:** Open and effective communication channels facilitate information flow between team members, stakeholders, and other relevant parties, ensuring everyone is on the same page regarding project progress, issues, and changes.
3	**Robust Risk Management:** Identifying, assessing, and managing risks throughout the project lifecycle helps in minimizing potential disruptions and ensures that the project stays on track.
4	**Resource Allocation:** Proper allocation of resources such as budget, personnel, equipment, and time is crucial for meeting project requirements within constraints.
5	**Roles and Responsibilities:** Clear definition of roles and responsibilities helps in avoiding confusion and ensures that each team member knows what is expected of them.
6	**Management Procedures:** Establishing procedures for managing changes to project scope, schedule, and budget helps in maintaining control over project outcomes and addressing unforeseen circumstances effectively.
7	**Quality Assurance:** Implementing quality assurance processes ensures that project deliverables meet predefined quality standards and satisfy stakeholder expectations.
8	**Performance Monitoring:** Regular monitoring of project performance against predefined metrics and benchmarks enables timely identification of deviations and facilitates corrective actions to keep the project on track.
9	**Documentation**: Having documented procedures and proper documentation management practices ensure that project information, decisions, and lessons learned are captured and accessible for future reference.
10	**Stakeholder Engagement:** Engaging stakeholders throughout the project lifecycle and managing their expectations is crucial for ensuring alignment with project goals and securing support
11	**Compliance and Governance:** Adhering to relevant regulations, standards, and organizational policies ensures that the project operates within legal and ethical boundaries and mitigates potential risks associated with non-compliance.or project success.
12	**Knowledge Management:** Establishing mechanisms for capturing and sharing project knowledge and lessons learned promotes continuous improvement and enhances organizational learning.

SOFTWARE PROCESS NAME: SCRUM

SOFTWARE PROCESS DESCRIPTION:

Scrum is an Agile framework that revolves around short, time-boxed iterations called *sprints*. Each sprint typically lasts 1-4 weeks and begins with sprint planning, where the team selects items from the product backlog to work on.
During the sprint, the team works on the selected items, holding daily stand-up meetings to track progress.
At the end of the sprint, the team presents the completed work to stakeholders during the sprint review, gathers feedback, and updates the product backlog.
The sprint concludes with a retrospective meeting where the team reflects on the sprint and identifies areas for improvement. This cycle of planning, executing, reviewing, and reflecting is repeated in subsequent sprints, allowing for incremental delivery and continuous improvement.

SOFTWARE PROCESS MODEL:

- The project begins with a comprehensive analysis of requirements and user needs, identifying key features and functionalities for the recommendation application. The product backlog is created, encompassing tasks related to web scraping, database setup, UI/UX design, machine learning model integration, and security implementation.
- **Sprint Planning:** The Scrum involves dividing the work into specific time periods called "Sprints." Before each sprint, a sprint planning session is held where the team selects user stories from the product backlog to work on during the sprint. Tasks are estimated, and sprint goals are defined the team collaborates to select work items and set goals.
- **Sprint**: A sprint is a time-boxed period, typically lasting 2-4 weeks. During the sprint, development work takes place according to the sprint plan, with daily stand-up meetings to discuss progress, challenges, and adjustments. Tasks are implemented, including setting up web scraping scripts, designing database schemas, developing user interfaces, integrating machine learning algorithms, and implementing security measures.
- **Sprint Review:** At the end of the sprint, the team conducts a review meeting to showcase the completed increment to stakeholders. This allows the product owner and other stakeholders to evaluate progress on the product and provide feedback.
- **Sprint Retrospective:** Following the review meeting, the team holds a retrospective to reflect on their process. During this meeting, team members discuss what went well, what could be improved, and how they can work more effectively in the future. This continuous improvement mindset helps the team refine their practices and optimize their workflow.
- **Incremental Delivery:** Throughout the project, working increments of the recommendation application are delivered at the end of each sprint, allowing for early validation of features and continuous improvement based on feedback.
- **Integration and Testing:** Development and testing occur concurrently throughout the project, with automated testing processes integrated into the development pipeline to ensure product quality and reliability.

REASONS TO CHOOSE THIS MODEL:

1. **Flexibility and Adaptability:** The domain of our application is characterized by rapidly evolving user preferences and content landscapes. Scrum's iterative and incremental approach allows for flexibility in accommodating changing requirements and adapting to emerging trends. This ensures that the application remains responsive to user needs and competitive in the market.

2. **Incremental Delivery:** Scrum emphasizes delivering working increments of the product at the end of each sprint. For our application, this means that users can start benefiting from personalized recommendations early in the development process. Incremental delivery enables continuous feedback from users, allowing the team to validate features, refine algorithms, and make necessary adjustments based on real-world usage.

3. **Continuous Improvement:** The sprint review and retrospective meetings in Scrum promote a culture of continuous improvement. This is particularly beneficial for our application, where the accuracy and relevance of recommendations play a critical role in user satisfaction. By regularly reviewing user feedback, analyzing usage patterns, and reflecting on development processes, the team can continuously enhance the application's recommendation algorithms and user experience.

4. **Risk Management:** Scrum's iterative approach allows for early detection and mitigation of risks. By delivering working increments of the application regularly, the team can identify potential issues, validate assumptions, and course-correct as needed. This proactive risk management approach reduces the likelihood of major setbacks and increases the project's overall success rate.

5. **Customer Satisfaction:** The iterative nature of Scrum ensures that our application evolves iteratively based on user feedback and changing market dynamics. By delivering value incrementally and continuously improving the application, Scrum maximizes customer satisfaction and retention, leading to long-term success and user engagement.