SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE PROCESS MODEL DOCUMENT

PROJECT NAME: CodeCraft

GROUP MEMBERS: Mert Koçyiğit, Haktan Yolcu, Mehmet Taşoğlu, Başar Kocabaş, Lorin Erol, Efran Ergelen

#	NECESSARY NEEDS FROM THE ORGANIZATIONAL PROCESS
1	Utilize a project management tool (e.g., Asana, Trello) to effectively manage tasks, track progress, and maintain clear communication within the development team.
2	Establish clear communication channels and protocols for efficient information sharing and collaboration between team members (developers, designers, educators).
3	Implement processes for effectively transferring knowledge and best practices within the development team. This could involve workshops, mentoring programs, or a centralized knowledge repository.
4	Develop a plan for managing communication and expectations with key stakeholders (e.g., investors, educators, potential users). This could involve regular meetings, progress reports, and feedback mechanisms.
5	Establish clear guidelines for content creators (educators, SMEs) regarding content format, quality standards, and alignment with learning objectives. Implement a quality control process to ensure content meets these standards.
6	Ensure the platform adheres to accessibility standards (e.g., WCAG) to cater to users with diverse needs.
7	Implement robust security protocols to safeguard user data (e.g., personal information, learning progress).
8	Develop a plan to market the platform and attract users. This could involve online advertising, social media campaigns, or partnerships with educational institutions.
9	Analyze user behavior and learning progress to identify areas for improvement and optimize the learning experience.

SOFTWARE PROCESS NAME: SCRUM
SOFTWARE PROCESS DESCRIPTION:

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE PROCESS MODEL DOCUMENT

Scrum is an iterative and incremental Agile methodology. Work is divided into short cycles called sprints (typically 1-4 weeks) where a subset of features from a prioritized list (product backlog) is tackled. Daily stand-up meetings ensure constant communication and collaboration. After each sprint, functionalities are reviewed and feedback incorporated into the next iteration. This allows for continuous progress, ensures the platform caters to user needs, and enables adaptation to the ever-changing learning landscape.

SOFTWARE PROCESS MODEL:

Sprint 1: Building the Foundation (Core User Management & Interface)

Focus: This sprint lays the groundwork for user interaction and platform navigation.
User Stories:
Users can create accounts and log in securely.
Users can browse available learning materials and search for specific topics.
Users can manage their profiles and track their learning progress.
Deliverables:
Functional user registration and login system with secure authentication protocols.
Basic platform functionalities like search, navigation bar, and user profile management features.
Initial design mockups and prototypes for a user-friendly and intuitive interface.

Sprint 2: Learning Management System & Core Content Integration

Focus: This sprint establishes the core functionalities for delivering educational content and user interaction.
User Stories:
Users can access a structured learning path with different difficulty levels.
Users can explore a variety of learning materials like written tutorials and video lectures.

Users can attempt interactive coding exercises and receive immediate feedback on their attempts.

Deliverables:

A functional learning management system for organizing and delivering diverse learning content.

Initial set of high-quality learning materials (tutorials, video lectures) covering fundamental programming concepts.

Interactive coding exercises with immediate feedback mechanisms (correctness checks, hints, and explanations).

Content creation tools allowing educators and subject-matter experts to easily upload and manage various learning materials.

Sprint 3: Project-Based Learning & Introducing the AI Assistant

Focus: This sprint introduces project-based learning with initial AI assistant support.

User Stories:

Users can access small-scale coding projects with varying difficulty levels.

Users can view clear project requirements and expected functionalities.

Users can receive assistance from the AI assistant during project development (code completion suggestions, syntax error identification).

Deliverables:

Integration of small-scale coding projects with increasing difficulty levels.

Clear project descriptions, requirements, and expected functionalities for each project.

Basic AI assistant functionalities providing code completion suggestions and syntax error identification.

Project management tools within the platform for code editing, execution, and progress tracking.

Sprint 4: Expanding Learning Content & Refining the AI Assistant

Focus: This sprint focuses on enriching learning materials and enhancing the AI assistant's capabilities.

User Stories:

Users can access a wider range of learning materials catering to different programming languages.

Users can utilize a more comprehensive AI assistant providing explanations for complex concepts and debugging assistance.

Deliverables:

Introduction of initial support for popular programming languages (e.g., Python, Java).

Enhancement of AI assistant functionalities based on user feedback (e.g., providing explanations, debugging suggestions).

Content creation tools allowing for the development of language-specific learning materials.

Continued development and refinement of the AI assistant based on usage data and identified limitations.

Sprint 5: Gamification and Community Features

Focus: This sprint introduces elements of gamification and fosters a sense of community among users.

User Stories:

Users can earn badges and points for completing learning activities and projects.

Users can interact with each other through forums or discussion boards.

Users can provide feedback on learning materials and the platform itself.

Deliverables:

Implementation of a basic gamification system with badges and points to motivate users.

Integration of forum functionalities for user interaction and knowledge sharing.

User feedback mechanism to gather valuable insights and improve the platform continuously.

Sprint 6: Advanced AI Assistant Features and Platform Optimization

Focus: This sprint focuses on further enhancing the AI assistant's capabilities and platform performance.

User Stories:

Users can receive more in-depth explanations and debugging assistance from the AI assistant.

The platform performance is optimized for a smooth user experience.

Deliverables:

Advanced functionalities for the AI assistant, including code refactoring suggestions and personalized learning recommendations.

Performance optimization measures to ensure smooth platform operation and scalability for future growth.

REASONS TO CHOOSE THIS MODEL:

SE 216 – SOFTWARE PROJECT MANAGEMENT
SOFTWARE PROCESS MODEL DOCUMENT

1. Adaptable Learning Platform for Evolving Landscape:

Leveraging Scrum's iterative nature, CodeCraft can continuously adapt to the dynamic programming education landscape. Regular user testing throughout sprints allows for gathering real-time feedback. This enables the team to:

Address unforeseen challenges: Quickly adjust the platform based on emerging trends or unexpected user needs.
Integrate new features seamlessly: Respond swiftly to evolving learning styles and preferences by incorporating relevant functionalities within upcoming sprints.
2. Continuous Improvement Driven by User Insights:

Scrum's emphasis on user testing throughout development cycles aligns perfectly with CodeCraft's focus on user-centric development.

Early and frequent user feedback: Gathered during sprints allows for identifying usability issues and areas for improvement.
Data-driven decision making: Insights from user testing inform platform updates and ensure functionalities directly address user needs, minimizing the risk of features that miss the mark.
3. User-Centric Development with Prioritized Learning Impact:

Scrum prioritizes functionalities with the highest user value, directly benefiting learners. This aligns with CodeCraft's core objective:

Focus on user stories: Prioritize development of features with the most significant learning impact within each sprint.
Deliver functionalities that directly address user needs: Ensure the platform delivers features that contribute to users' programming skill development.
4. Reduced Risk and Faster Time-to-Market:

Scrum's iterative approach mitigates project risks associated with lengthy development cycles:

Early and continuous feedback: Allows for course correction and adjustments throughout the process, minimizing the chance of investing resources in features that might not resonate with users.

Incremental delivery: Frequent deployment of core functionalities in smaller batches enables faster time-to-market. This allows CodeCraft to capture early user feedback and establish a user base quicker.

5. Efficient Collaboration and Cost-Effectiveness:
Scrum fosters a collaborative development environment:
Daily stand-up meetings: Promote open communication and transparency within the team, ensuring everyone is aligned and aware of potential roadblocks.
Focus on delivering prioritized value: Reduces the risk of wasted resources as the team focuses on functionalities with the highest user impact.
Early identification and correction of issues: Minimizes rework and associated costs.