**이론 과제 #2**

**(1) Communication (의사소통)**

* **이해관계자(사용자, 고객, 팀원 등)와 소통하여 요구사항을 수집**합니다.
* 프로젝트의 목적, 기대 효과, 필수 기능, 제약 조건 등을 명확히 파악합니다.
* 회의, 인터뷰, 설문, 이메일 등 다양한 방법으로 정보를 수집하고, 요구사항을 문서화합니다.
* 요구사항이 모호하거나 충돌하는 부분이 있다면 조율하여 명확히 합니다.

**(2) Planning (계획수립)**

* **프로젝트 전체 일정과 마일스톤을 수립**합니다.
* 각 기능별/단계별로 작업 분배(누가, 언제, 무엇을 할지)와 역할을 정합니다.
* 필요한 자원(인력, 예산, 도구 등)을 산정하고 확보 계획을 세웁니다.
* 예상되는 위험요소(리스크)를 식별하고, 대응 방안을 마련합니다.
* 개발, 테스트, 배포 등 각 단계의 시작/종료 시점을 계획합니다.

**(3) Modeling (모델링)**

**- Analysis of Requirements (요구사항 분석)**

* 수집한 요구사항을 정리하여 **기능적/비기능적 요구사항 명세서**를 작성합니다.
* 사용자 시나리오, 유스케이스, 데이터 흐름도 등으로 요구사항을 시각화합니다.
* 요구사항 간 우선순위를 정하고, 누락/중복/모순이 없는지 검토합니다.

**- Design (설계)**

* 시스템의 전체 구조(아키텍처)를 설계합니다.
* 주요 모듈, 데이터베이스, 인터페이스(UI/UX), 데이터 흐름 등을 구체적으로 설계합니다.
* 설계 산출물(클래스 다이어그램, 시퀀스 다이어그램, ERD 등)을 작성합니다.
* 성능, 보안, 확장성 등 비기능 요구사항을 설계에 반영합니다.

**(4) Construction (구현)**

**- Code Generation (코드 작성)**

* 설계서를 바탕으로 실제 소스코드를 작성합니다.
* 모듈별로 개발하고, 코드 리뷰를 통해 품질을 점검합니다.
* 버전 관리 시스템(Git 등)을 활용해 코드 변경 이력을 관리합니다.

**- Testing (테스트)**

* 단위 테스트, 통합 테스트, 시스템 테스트 등 다양한 테스트를 수행합니다.
* 테스트 케이스를 작성하고, 실제로 실행하여 버그를 찾고 수정합니다.
* 요구사항이 모두 충족되는지, 성능/보안 등 비기능 요구도 검증합니다.

**(5) Deployment (배포)**

* 개발이 완료된 소프트웨어를 실제 운영 환경에 배포합니다.
* 설치, 설정, 데이터 마이그레이션 등 배포에 필요한 작업을 수행합니다.
* 사용자 매뉴얼, 설치 가이드, FAQ 등 문서를 제공합니다.
* 사용자 교육, 초기 지원, 피드백 수집을 통해 안정적으로 운영될 수 있도록 지원합니다.
* 배포 후에도 유지보수 및 업데이트 계획을 세웁니다.