

Software Development
Project



SanHark Project

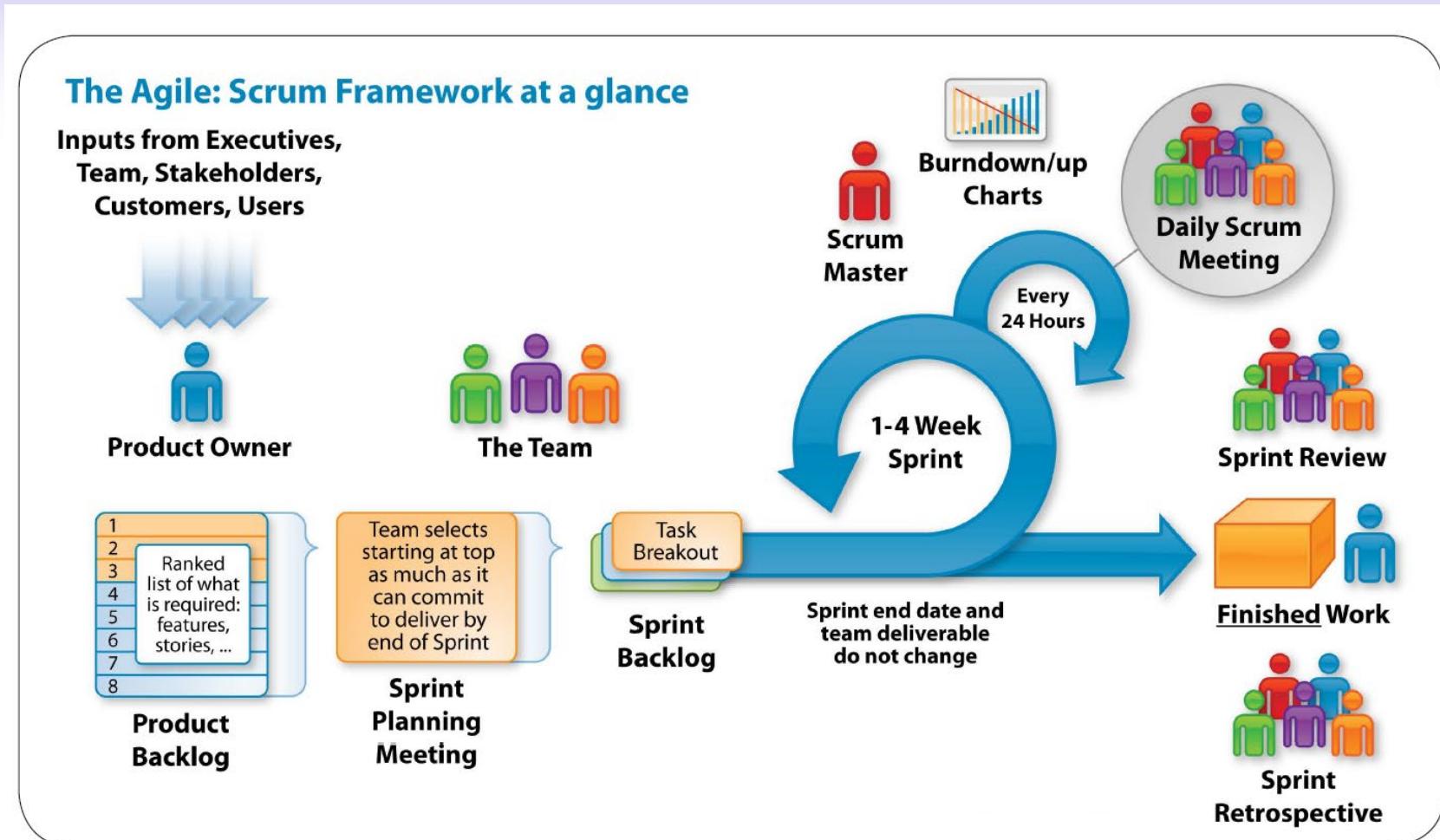
Project Process

- Agile Approach

Fall, 2025

Jang-Eui Hong

Scrum Process



프로젝트 참여자 역할

프로덕트 오너 (Product Owner)

- 소프트웨어 개발의 주요 책임자
- 요구사항의 결정, 변경, 관리에 대한 책임을 가짐

스크럼 마스터 (Scrum Master)

- 스크럼의 리더
- 스크럼에 대한 진행 및 통합 관리, 기술적 문제 해결
- 프로젝트 리스크 관리

팀원 (The Team) > TI (Tech. Info.) 메니저

- 구현 담당
- 개발 관련 기술 정보 통합 및 유지 관리
- 프로덕트 오너와 스크럼 마스터도 팀원으로 참여

단계 1: 시스템 정의서 개발

시스템 정의서

- 프로젝트에서 수행할 소프트웨어 시스템에 대한 간단한 설명서

시스템 정의서 포함 내용

- 시스템 명칭
- 시스템에 대한 간략한 설명
- 시스템 개발의 목적 (OOO의 편의성 향상)
- 시스템 운영 환경 및 사용자 범주
- 시스템이 제공하는 핵심 기능 (최소 5개 이상)과 간단한 설명
- 기존에 존재하는 시스템(혹은 소프트웨어)과의 비교 분석



시스템 정의서

단계 2: 프로젝트 계획서 작성

프로젝트 계획서

- 프로젝트 수행 기간에 대한 일정, 팀 구성 및 역할, 개발 산출물 등에 대하여 설명하는 문서

(변형된) 프로젝트 계획서 구성

- 시스템 명칭 (1줄 설명 포함)
- 팀구성 및 팀원의 역할
- 프로젝트개발 및 운영 환경 정의
 - 요구사항 정의, 리스크 정의 : Excell
 - 미팅 방법 : 온라인/오프라인 (카톡, 줌)
 - 사용할 개발 도구(IDE), Git, 기술
 - 기타 문서 : 한글
- 주요 개발 산출물 목록

시스템 정의서 (1)
프로젝트 계획서 (1)
시스템 아키텍처 (1)
Product Backlog (1)
Sprint Backlog (N)
Technical Spec (N)
Risk Register (1)
Burndown Chart (0)
통합 Test Result (1)



프로젝트 계획서 (2~3p)

단계 3: 아키텍처 스케치 (Sketch)

시스템 아키텍처

- 시스템 정의서를 기반으로, 상위 수준의 사용자 요구 기능을 표현하는 전체 시스템의 구조

아키텍처 설계 가이드라인

- 시스템 정의서를 기반으로 핵심 액터(Actor)를 식별하고,
- 액터별로 요구되는 시스템 기능을 도출
- 최상위 수준 (주제)에서의 모듈화된 기능 식별
- Use Case Diagram을 이용하여 Big Picture 도식화
 - 상위 수준에서 모든 기능이 포함되도록 정의해야 함.
 - Use Case 설명서는 작성하지 않으나, Use Case 목록표는 작성함. (UC 이름 + 간단한 설명)



시스템 아키텍처

단계 4: Product Backlog 개발

제품 백로그(Product Backlog)

- 개발 대상 시스템에서 제공하고자 하는 기능의 정의 및 우선순위 목록

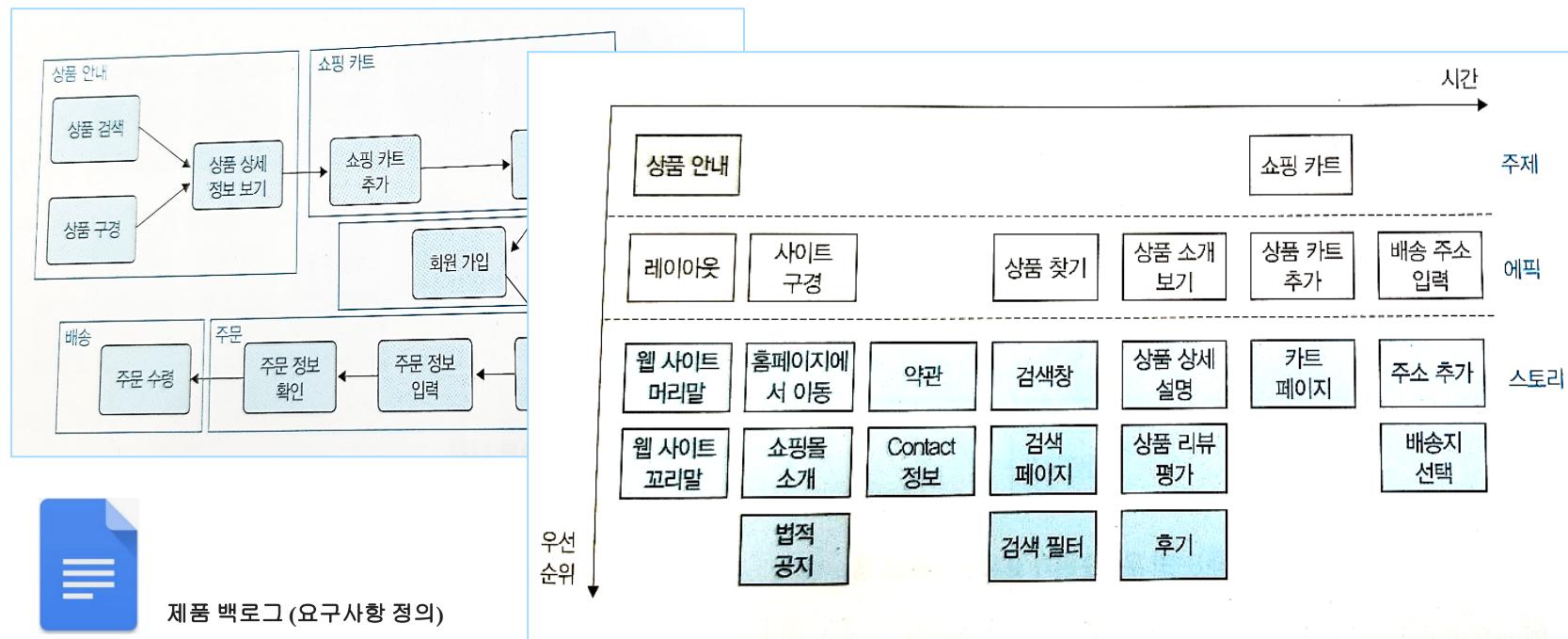
제품 백로그 작성 가이드라인

- (1) 시스템 정의서와 Use Case 다이어그램을 이해한다.
- (2) 팀원 각자 유저 스토리(User Story)를 식별, 정의 한다.
- (3) 식별된 유저 스토리를 통합하고, 시간순으로 나열한다.
- (4) 유저 스토리를 그룹핑하여 주제(서브 시스템) 단위로 분류한다.
- (5) 시스템 – 주제 – 에픽(기본 기능) – 스토리 순으로 정리한다.
- (6) 주제, 에픽, 스토리에 대하여 각각 우선순위를 부여한다.
 - 중요도, 구현 능력 등을 고려하여 부여함.
 - 고유한 우선순위 : 에픽 단위 부여 / 스토리 단위 : 에픽 순위-n

단계 4: Product Backlog 개발

제품 백로그(Product Backlog) 문서화

- 확정된 유저 스토리를 종합하여 관리한다. (엑셀 이용)
 - 주제+에픽+스토리명+사용자+스토리+우선순위+추정 작업량 +진도
- 제품 백로그는 추후 변경 가능하며, 지속적인 관리가 필요함.



단계 4-1: 작업량 추정(Planning Poker)

스토리 포인트 기준점 선정

- 제품 백로그 목록중에서 팀원 모두가 이해하는 기준 항목 결정
- 기준 항목을 기능 개발을 위해 1~2주내로 마무리할 수 있는 작업

Planning Poker

- 기준 항목의 규모는 Face Value == 3으로 지정
- 모두 항목(스토리)을 하나씩 살펴가면서 스토리 포인트 결정
 - 기준이 되는 기능 규모보다 크면(2배 수준) Face Value == 5
 - 기준 규모보다 절반 수준에 해당된다면 Face Value == 1 or 2
- 팀원이 각 항목에 대하여 1, 2, 3, 5의 값을 갖는 포커를 제시
 - 민장 일치 : 해당 값으로 스토리 포인트를 결정
 - 만장일치 아닌 경우 : 각자 이유를 설명하고 포커를 재실시

단계 4-1: 작업량 추정(Planning Poker)

제품 백로그의 간신

- Planning Poker를 통해 정해진 스토리 포인트를 제품 백로그 목록에 기재
- 만약 요구사항의 스토리 포인트가 5이상인 것이 있다면
 - 요구사항을 스토리 포인트 3점 기준에 적합하도록 세분화
- 너무 작은 스토리 포인트 (1)의 경우 통합 가능 여부 판단
 - 통합은 동일한 기능 그룹내에서 진행하며 Coupling 이 높은 경우 통합



(간신된 제품백로그)

단계 5: Sprint Planning

스프린트 (Sprint)

- 달력을 기준으로 하는 1주 ~ 4주 단위의 개발 기간

스프린트 계획 미팅 수행

- 갱신된 제품 백로그 목록에서 개발 대상 작업 선정
 - 스토리 단위로 선정하되, 스토리의 그룹(group)도 가능
 - 개발 작업 선정시 고려사항
 - : 사전 정의된 우선순위 + 개발팀의 기술 역량 + 기능 중요도
- 선정 작업에 대한 검토
 - 스프린트 선정 작업에 대한 중요도, 가시성, 설명성 등 고려
 - 구현 기술에 대한 진단을 통해 구현 가능성 반드시 확인

→ 스프린트 백로그 작성 진행

단계 6: 스프린트 백로그 작성

스프린트 백로그(Sprint Backlog)

- 하나의 스프린트 동안 개발할 목록으로, 요구 기능을 개발, 완료하기 위한 작업을 세부 태스크로 정의
- 별도의 엑셀 파일로 작성하며, 하나의 쉬트에 하나의 스프린트 백로그 항목을 배정
- 각 태스크의 크기는 시간 단위로 추정

하나의 스프린트 백로그를 완성하기 위한 가이드라인

- 식별된 태스트에 대한 개발 담당자 및 소요 시간 지정
- 스프린트 백로그의 완료 기준 설정 및 팀원 합의
 - 기능 정확도 + 담당자 + 종료일 등을 합의

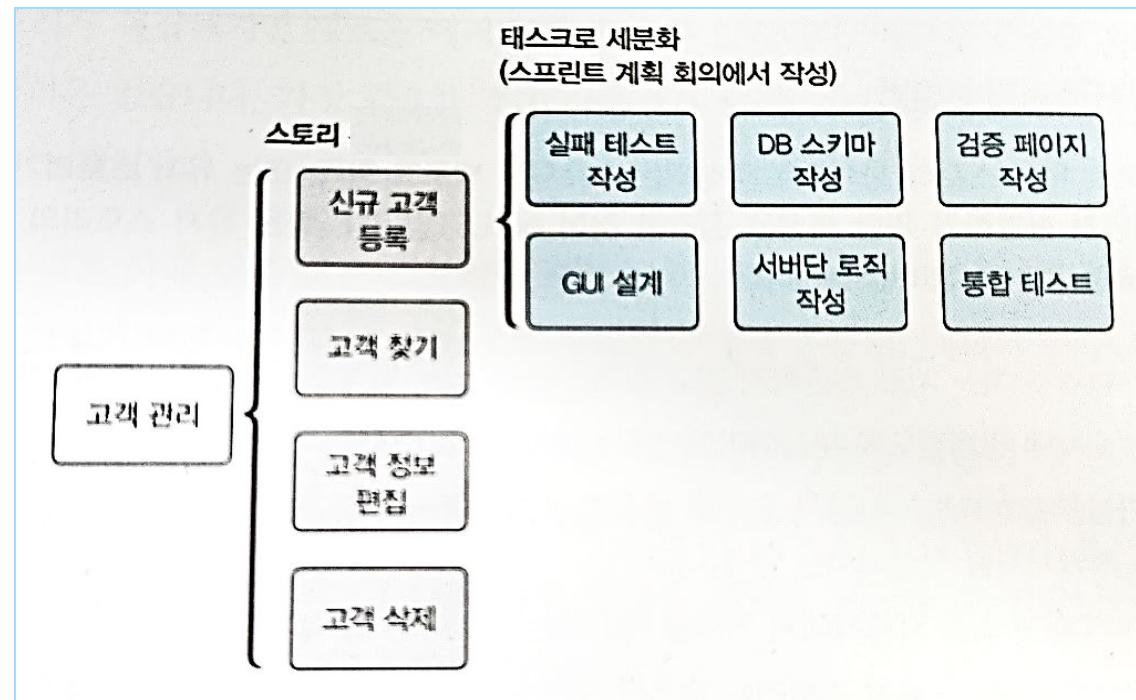
단계 6: 스프린트 백로그 작성

스프린트 백로그(Sprint Backlog)

- 스프린트 주기 동안에 수행해야 하는 작업 목록임
- 엑셀 = {스프린트 ID + 에픽(스토리) ID + 태스크 ID + 태스크명 (설명)
+ 소요 기술 + 담당자 + 종료일 + 진도}



{스프린트 백로그 – SB(n)}



단계 7: 스프린트 운영

사전 조건 : 개발 프로젝트 진행을 위한 환경 구축(예: GIT)

스프린트 백로그에 나타난 작업을 개인별로 수행.

만약 짹 프로그래밍 방법을 사용한다면

- 2명이 1조가 되어 프로그램 개발 진행 가능함.

스프린트 운영동안의 세부 활동들

- (1) 팀원에게 부여된 단위 태스크 업무 개발 시작
- (2) Daily Meeting을 통한 문제 해결 및 진도 점검
- 리스크 레지스터 및 번다운 채트 검토
- (3) 개발된 코드에 대한 테스트 수행
- (4) 지속적인 통합 활동 수행
- (5) 간결한 설계 활동 수행

단계 7-1: 단위 태스크 수행 (구현)

스프린트 백로그에 나타난 태스트 단위 구현

- 구현 언어와 개발 환경에 대한 준비 여부 점검
- 구현 과정에서 지켜야 하는 기본 규칙을 준수하여 진행
- 추후 확장성 및 통합 용이성을 고려한 구조화된 구현
- 필요시 생성형 AI 도구의 도움을 받아 진행 가능

산학 프로젝트 교과에서의 적용 가이드라인

- 각자에게 주어진 태스크에 대하여 자신의 스케줄 관리 철저
- 스크럼 마스터는 스크럼 운영에 대한 전반적인 조정 및 관리



{구현 산출물 – 코딩 등 }

단계 7-2: Daily Meeting 수행

Daily Meeting은 팀원들이 매일 아침 수행하는 20분 미팅

- 앓지 않고 서서하는 미팅
- 어제의 개발 내용과 개발 과정의 문제점 및 공유할 사항 토의
- 절대 20분을 넘기지 말며, 모든 사람이 돌아가면서 얘기 하기
- 현재까지의 팀 전체 진행 작업에 진도 점검과 번다운 챕트 간신
 - 수행 작업에 대한 진도율 표현

산학 프로젝트 교과에서의 적용 가이드라인

- 적어도 2~3일에 한번씩은 Daily Meeting을 수행
- 온라인 회의로 진행 가능
 - 줌 권장 – 회의 내용 저장 관리
 - 카톡 활용 가능 – 카톡 내용 저장 관리



(간신된 번다운 챕트)

단계 7-3: 리스크 레지스터 작성 관리

리스크 레지스터

- 스프린트 태스크 수행동안 발생하는 리스크를 정리하고 관리하기 위한 문서

리스크 레지스터 구성 및 활동

- Daily Meeting 과정에서 논의되는 리스크를 확정 관리
- 리스크 관리 가이드라인 (엑셀)
 - 리스크 ID + 스프린트 ID + 태스크 ID + 리스크 내용 + 해결 (완화) 방법 + 해결 종료일 + 담당자
- 다음 Daily Meeting에서 리스크 관리 상태 및 처리 결과 공유



리스크 레지스터

단계 7-4: 단위 테스트 수행

현재 활용하는 도구를 이용하여 테스트 수행

- 계획된 모든 세부 기능이 구현되었는가?
- 구현된 기능이 정확하게 수행되는가?

테스트 진행 가이드라인

- 단위 테스트 수행을 위해, 테스트 사례(Test Case) 설계
- 테스트를 수행하여, 결과 점검 및 결함 수정
- 테스트 수행을 위한 드라이버, 스텁 개발(필요시)
- 모든 테스트 사례가 오류 없이 통과될 때 까지 진행

테스트 수행을 위한 지원도구(예: Junit, pytest) 활용 추천



(테스트 결과서)

단계 7-5: 스프린트 설계 활동

스프린트 마감 전에 수행하는 간단한 설계 활동

한글과 같은 워드를 이용하여 작성 권장

- 문서 식별자 : Project 명칭-S1_TS_xxxx_v1)

설계 활동 주요 대상

- DB 스키마 설계 내용
 - 테이블명 + 용도 + 필드명 + 데이터 타입 + 주키/보조키
- 클래스 명세
 - 클래스명 + 속성명 + 메소드명 + 간단한 설명
- API 명세 (필요시) : 외부 인터페이스 대상
 - 인터페이스명 + 클래스명 + 시그너처 + 설명



{스프린트 기술 명세}

단계 7-6: 지속적인 통합

지속적인 통합

- 팀이 작업한 소스코드, 데이터베이스 스크립트, 코드검사(Inspection), 자동화 테스트 등을 가능하면 자주 통합하는 방향으로 진행
- 최소 하루에 한번이상, 여러번 통합을 진행

지속적 통합을 위한 환경 구축

- GIT을 이용한 지속적 통합 수행
- SVN, Maven 등을 활용한 버전 관리 및 빌드 관리 수행



통합 결과물
(GIT 저장소)

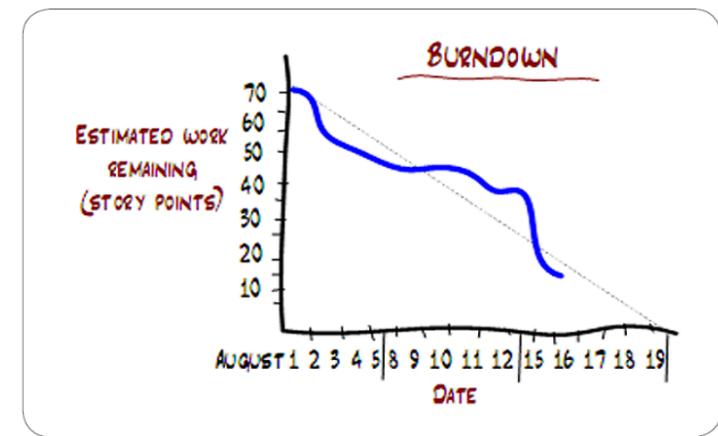
단계 7-7: 번다운 챕트(Burndown Chart)

번다운 챕트

- 개발을 완료하기까지 남은 작업량을 보여주는 그래프
- 각 스프린트 별로 남아있는 작업량을 스토리 포인트 값으로 표현

2가지로 구별하여 유지 관리

- 스프린트 백로그 번다운 챕트
- 제품 백로그 번다운 챕트



(변형된) 번다운 챕트 관리방법

- 제품 백로그의 스토리 단위 진도를 마킹
- 스프린트 백로그에 태스크 단위 진도를 마킹

단계 8: 스프린트 리뷰

스프린트 리뷰

- 하나의 스프린트가 종료되었을 때, 결과물이 요구사항을 충족하는지 검토하는 활동
- 사용자와 함께 개발된 결과물을 데모하고, 사용자 피드백 수집
- 체크리스트를 정의하고, 이를 활용하여 점검 수행
- 추후 교과 시간에 담당 교수에게 스피린트 결과물 발표 필요

스프린트 리뷰 수행의 사전 조건

- 스프린트 백로그에 정의된 모든 태스크의 종료 점검
- 스프린트 백로그의 완료 기준 충족 여부



{스프린트 리뷰 회의록 – 체크리스트 형식}

단계 9: 회고(Retrospective) 미팅

리트로스펙티브 미팅

- 각 스프린트가 종료된 뒤, 팀이 프로세스를 개선하고 추후 동일한 문제점을 반복하지 않기 위해 실시하는 회의

리트로스펙티브 미팅 가이드라인

- 스크럼 마스터가 스프린트 수행에 대한 간단한 설명과 개발 결과를 정리
- 각 구성원(member)은 잘한 일, 아쉬운 일 등을 메모지에 작성
- 내용을 취합하여, 개선할 점을 찾고, 개선 방안 토의 하기

스프린트 리뷰와 함께 실시할 수 있음.



(회고 미팅 회의록)

단계 10 : 반복 개발

단계 5번부터 단계 9번까지 반복적으로 수행

- 각 스프린트는 1주에서 3주 단위로 결정하되,
- 스프린트 백로그에서 식별된 태스크 수행 시간을 고려하여 결정

매 스프린트 종료시점에서 생성되는 문서 관리

- 리트로스펙티브 미팅이 종료 시점
- 하나의 스프린트내에서 작성된 문서를 디렉토리로 관리
- 문서 작성은 다양한 도구(엑셀, 워드, CASE 툴 등을 이용하여 생성)

스프린트 단위 개발 산출물 등은 GIT에 저장 관리

단계 11: 개발 통합 및 시나리오 테스트

모든 스프린트가 종료되는 시점

- 제품 백로그 번다운 차트 확인 (프로덕트 오너)
 - 제품 백로그에서 식별된 모든 작업이 완료되었는가?
- GIT에 저장된 코드의 최종 버전 통합 및 확인 (스크럼 마스터)
- 통합 테스트 수행
 - 제품 백로그의 각 항목별로 시나리오 기반 테스트 수행
 - 필요시 오류 수정 및 기술 정보 명세 갱신
- 스프트린단위 산출물에 대한 문서화 통합 (IT 메니저)
 - 스프린트 백로그 / 도메인 모델 / 테스트 결과 등의 통합 작성



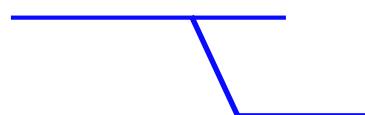
통합 테스트 결과서

단계 12: 프로젝트 종료

프로젝트 종료 시점에서의 산출물 목록

1. 시스템 정의서 (1~2 페이지)
2. 프로젝트 계획서 (1~2페이지)
3. 시스템 아키텍처 (1~2 페이지)
4. 최종 제품 백로그 (Product Backlog) : 엑셀 파일
5. 스프린트 단위별 문서 집합

6. 통합 테스크 결과서
7. GIT에 저장된 개발 산출물

- 
- 스프린트 백로그 : 엑셀 파일
 - 스프린트 기술 명세 : 한글 파일
 - 회의록
 - : Daily meeting,
 - : 스프린트 리뷰 체크리스트

* Burndown 차트는 백로그의 내용으로 대체

산학 프로젝트 – 프로젝트 프로세스

질의 및 응답

