

프로젝트 계획서

<준식쓰와 형님들>

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

- 변 경 이 력 -

일자	버전	변경 내역	작 성 자
2019. 11. 27.	Ver1.0.0	초안 작성	모두

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

- 목 차 -

1.	프로젝트 개요	- 3 -
1.1	목적	- 3 -
1.2	주요 일정	- 3 -
1.3	조직	- 4 -
1.3.1	조직도	- 4 -
1.3.2	역할 및 책임	- 4 -
1.4	생명주기 모델 : 애자일[AGILE] 방법론	- 5 -
1.5	도구	- 6 -
2.	규모 산정	- 7 -
2.1	WBS(WORK BREAKDOWN STRUCTURE)	- 7 -
3.	일정	- 8 -
4.	산출물 관리(팀원들 간의 관리 계획).....	- 9 -
5.	리스크 관리 계획	- 10 -
5.1	프로젝트 측면 위험	- 10 -
5.2	기술적 측면 위험	- 10 -
5.3	사용자 측면 위험	- 11 -

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

1. 프로젝트 개요

1.1 목적

- 프로젝트명: Penta-Go
- 프로젝트 기간 : 2019.11.26. ~ 2019.12.09.
- 개발 목적
 - 펜타고는 스웨덴에서 개발된 고차원 오목게임이다. 중국에서 유래된 기존의 오목과 달리, 구슬판 4개를 오른쪽 또는 왼쪽으로 90도 회전시키며 게임이 진행된다.
 - 따라서 다양한 변수가 있고 입체적인 생각을 유도하고 있어 높은 차원의 사고를 할 수 있게 도와준다. 2006년 미국 멘사에서 선정한 게임이기도 하며 프로그래밍을 함에 있어 필요한 유연한 사고력을 증진시키기 위해 컴퓨터 과학과 학생들이 즐길 수 있도록 해당 게임을 구현해보기로 하였다.

1.2 주요 일정

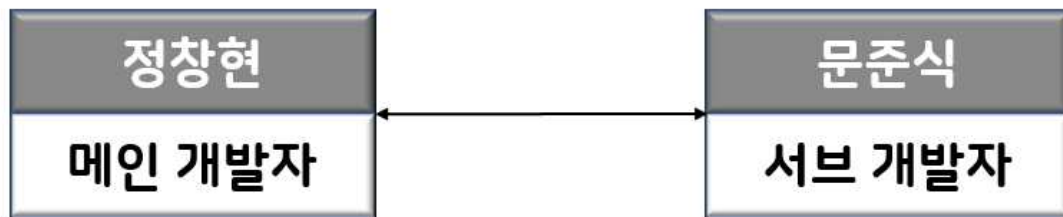
단계	일정	산출물
요구사항 명세	2019.11.26. ~ 2019.11.26.	요구사항 명세서
계획	2019.11.27. ~ 2019.11.28.	프로젝트 계획서
구현	2019.12.29. ~ 2019.12.04.	소스 산출물
테스트	2019.12.05. ~ 2019.12.07.	

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

1.3 조직

1.3.1 조직도

수평적 조직



1.3.2 역할 및 책임

팀원	역할	책임
정창현	메인 개발	프로그램 개발 (메인)
문준식	서브 개발 및 문서	프로그램 개발 (서브) 및 문서 산출물 제작

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

1.4 생명주기 모델 : 애자일[AGILE] 방법론

애자일 방법론은 소프트웨어 개발 방법에 있어서 아무런 계획이 없는 개발 방법과 계획이 지나치게 많은 개발 방법들 사이에서 타협점을 찾고자 하는 방법론이다. 계획이 없는 방법론의 경우, 앞으로의 일을 예측하기 힘들고 효율적이지 못하다는 점에서 취약점을 가지고 있으며, 계획에 너무 의존하는 경우는 그 형식적인 절차를 따르는데 필요한 시간과 비용을 무시할 수 없으며, 전체적인 개발의 흐름 자체를 느리게 하는 단점을 가지고 있다.

애자일 방법론이 다른 고전적인 방법론(예를 들면 폭포수 모델 또는 나선 모형)과 구별되는 가장 큰 차이점은 less document-oriented이다. 즉, 문서를 통한 개발 방법이 아니라, code-oriented, 실질적인 코딩을 통한 방법론이라는 점이다.

그러므로 애자일 개발 방법론은 계획을 통해서 주도해 나갔던 과거의 방법론과는 다르게 앞을 예측하며 개발을 하지 않고, 일정한 주기를 가지고 끊임없이 프로토타입을 만들어내며 그때그때 필요한 요구를 더하고 수정하여 하나의 커다란 소프트웨어를 개발해 나가는 adaptive style 이라고 할 수 있다.

애자일 방법론의 장점


- 1.프로젝트 진행 중간 중간에 필요한 요소들을 바꿀 수 있다.
- 2.시작할 때 프로젝트를 정확하게 규정하지 않아도 된다.
- 3.작은 요소들을 출시 할 때 빠르게 만들 수 있다.
- 4.점진적으로 테스트되기 때문에 초기에 버그를 발견할 수 있다.




프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

1.5 도구

 **VirtualBox** ✓Trello: UNIX 환경 세팅을 위한 가상머신

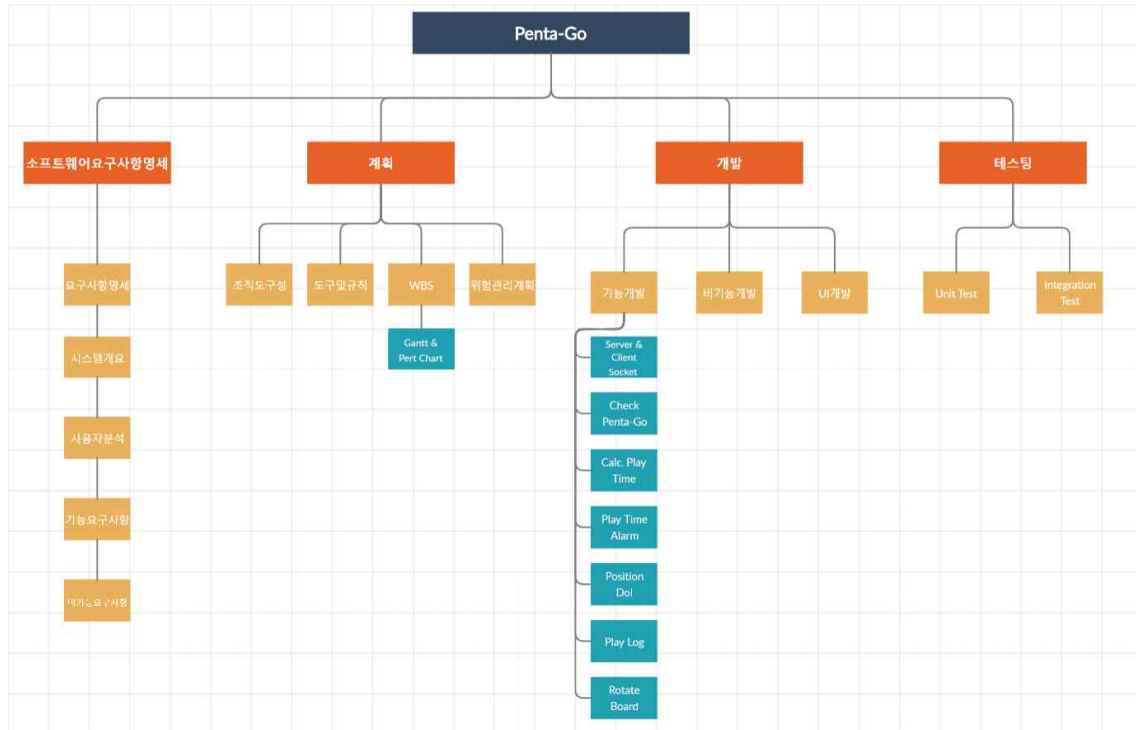
 **GitHub** ✓Github: 산출물 관리를 위한 분산 버전 관리 시스템

 **Office** ✓Microsoft Office: 문서 산출물 제작

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

2. 규모 산정

2.1 WBS(Work Breakdown Structure)



프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

3. 일정

분류	세부계획	2019.11.							2019.12.							2019.12.						
		24/일	25/월	26/화	27/수	28/목	29/금	30/토	1/일	2/월	3/화	4/수	5/목	6/금	7/토	8/일	9/월	10/화	11/수	12/목	13/금	14/토
인사관리 개발	요구사항명세서 작성																					
	프로젝트계획서 작성																					
	펜타고 프로그램 개발																					
	테스트																					
	프로젝트결과보고서 작성																					

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

4. 산출물 관리(팀원들 간의 관리 계획)

산출물 저장 방법

- 문서 산출물: Github에 저장, word 및 한컴오피스 한글을 이용하여 문서 관리
 - 요구사항 명세서, 프로젝트 계획서, 결과 보고서 등
- 소스 산출물: Github에 저장
 - Penta-Go 소스 코드 파일

산출물 명 규칙

- 문서 산출물: [PG]문서명_TeamName
 - (예: [PG]요구사항명세서_준식쓰와형님들.hwp)

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

5. 리스크 관리 계획

5.1 프로젝트 측면 위험

위험 요소	가능성	영향도	대처 방안
요구사항 변경	상	상	- 변경내용이 타당한가 검토 후 변경 - 지속적인 회의 및 의사소통을 유지하여 발생 최소화
팀원 역할 변경	하	중	- 상호 간 협력 유지 - 문서전달과 적극회의를 통한 인수인계
팀원 의견 충돌	상	중	- PM주도로 원만한 의견 중재안 도출 - 의견에 따른 근거 제시
일정 지연	중	상	- 개발관리대상 업무별 요구관리 철저 - Gantt, PERT 차트 항상 숙지
예산 초과	하	상	- 프로젝트 착수 시 정확한 비용 산정 - 주간, 월간 회의를 통한 철저한 예산 관리

5.2 기술적 측면 위험

위험 요소	가능성	영향도	대처 방안
기술력 부족	중	상	- 개발 인력에 대한 충분한 사전 교육 실시 - 개발 분야에 대한 구체적인 분석
성능 미달	중	상	- 시스템 구축 전 충분한 능력, 용량 산정
새로운 기술의 출현	하	중	- 프로젝트에 적용 가능한지 검토
산출물 관리 소홀	하	상	- 백업 및 복구 체제 마련 - 일일 백업 관리 철저
프로토타입의 테스트불가	중	상	- 개발 중 테스트 시점 미리 결정

프로젝트 명	Penta-Go	프로젝트 기간	2019.11.26.~2019.12.09.
문 서 명	프로젝트 계획서	버전	Ver 1.0.0

5.3 사용자 측면 위험

위험 요소	가능성	영향도	대처 방안
요구사항 누락	중	상	- 지속적 회의를 통한 체크리스트 확인
요구사항 변경	상	상	- 변경사항 상시 파악, 적용검토
수용가능 사용자 초과	하	중	- 예상 인원보다 많은 저장공간 확보 - 수용가능 인원수 재협의