**프로젝트 계획서**



<007>

**- 변 경 이 력 -**

|  |  |  |  |
| --- | --- | --- | --- |
| **일자** | **버전** | **변경 내역** | **작 성 자** |
| 2019.03.18 | 0.1v | 신규 (주요 일정, 조직 작성) | 신창호 |
| 2019.03.18 | 0.15v | 조직 역할 및 책임 수정 | 김형택 |
| 2019.03.20 | 0.2v | 프로젝트 개요 작성 | 김준기 |
| 2019.03.20 | 0.25v | 생명주기 모델 작성 | 김동민 |
| 2019.03.25 | 0.3v | 규모산정(wbs) 수정 | 김동민 |
| 2019.03.25 | 0.35v | 도구 작성 | 김형택 |
| 2019.03.27 | 0.4v | 생명 주기 모델 수정 | 오정엽 |
| 2019.03.28 | 0.5v | 일정 (Gantt 차트) 수정 | 오정엽 |
| 2019.03.28 | 0.55v | 생명 주기 모델 및 리스크 관리 계획 수정 | 이소현 |
| 2019.03.28 | 0.6v | 생명주기 모델의 적절성 내용 추가 | 김형택 |
| 2019.03.29 | 0.7v | 산출물 관리 수정 | 김준기 |
| 2019.03.30 | 0.75v | 산출물 관리 수정(slack내용 및 전반적 내용) | 김형택 |
| 2019.03.31 | 0.8v | 비용 산정(wbs) 및 주요 일정 수정 | 김형택 |
| 2019.03.31 | 0.85v | Pert 차트 추가 및 간트차트 수정 | 김동민 |
| 2019.04.01 | 0.9v | 주요 일정, 도구 수정 및 간트차트 수정 | 김형택 |
| 2019.04.01 | 1.0v | 최종 수정 (PPT 최종 검토, 가독성개선) | 김형택,김준기 김동민,이소현 |

- 목 차 -

[**1.**](#_3znysh7) **프로젝트 개요 - 3 -**

[1.1](#_3znysh7) 목적 - 3 -

[1.2](#_2et92p0) 주요 일정 - 3 -

[1.3](#_tyjcwt) 조직 - 4 -

[1.3.1](#_1t3h5sf) 조직도 - 4 -

[1.3.2](#_4d34og8) 역할 및 책임 - 4 -

[1.4](#_2s8eyo1) 생명주기 모델 - 5 -

[1.5](#_17dp8vu) 도구 - 5 -

[**2.**](#_3rdcrjn) **규모 산정 - 6 -**

[2.1](#_26in1rg) WBS(Work Breakdown Structure) - 6 -

[**3.**](#_lnxbz9) **일정 - 7 -**

[**4.**](#_35nkun2) **산출물 관리 - 8 -**

[**5.**](#_1ksv4uv) **리스크 관리 계획 - 9 -**

1. **프로젝트 개요**

* 1. **목적**

프로젝트 수행 시에 회사 구성원들 간의 효율적인 커뮤니케이션과 여러 명이 동시에 작업을 진행하기 위해 구글 API를 활용한 실시간 협업관리 도구 프로젝트를 기획 하였다.

회사는 여러 명이 함께 일을 하기 때문에 프로젝트를 수행하는 데 있어서 효율적인 커뮤니케이션을 이루는 것이 쉽지 않다. 그렇기 때문에 업무 중에 구성원 간의 커뮤니케이션이 원할해지면 업무 효율성이 증가하는 결과를 얻을 수 있다. 하지만 기업에서 사용하는 사내 커뮤니케이션 협업툴에 대해서는 많은 투자나 관심이 미치지 못했고 이는 업무의 효율성 저하를 초래하였다.

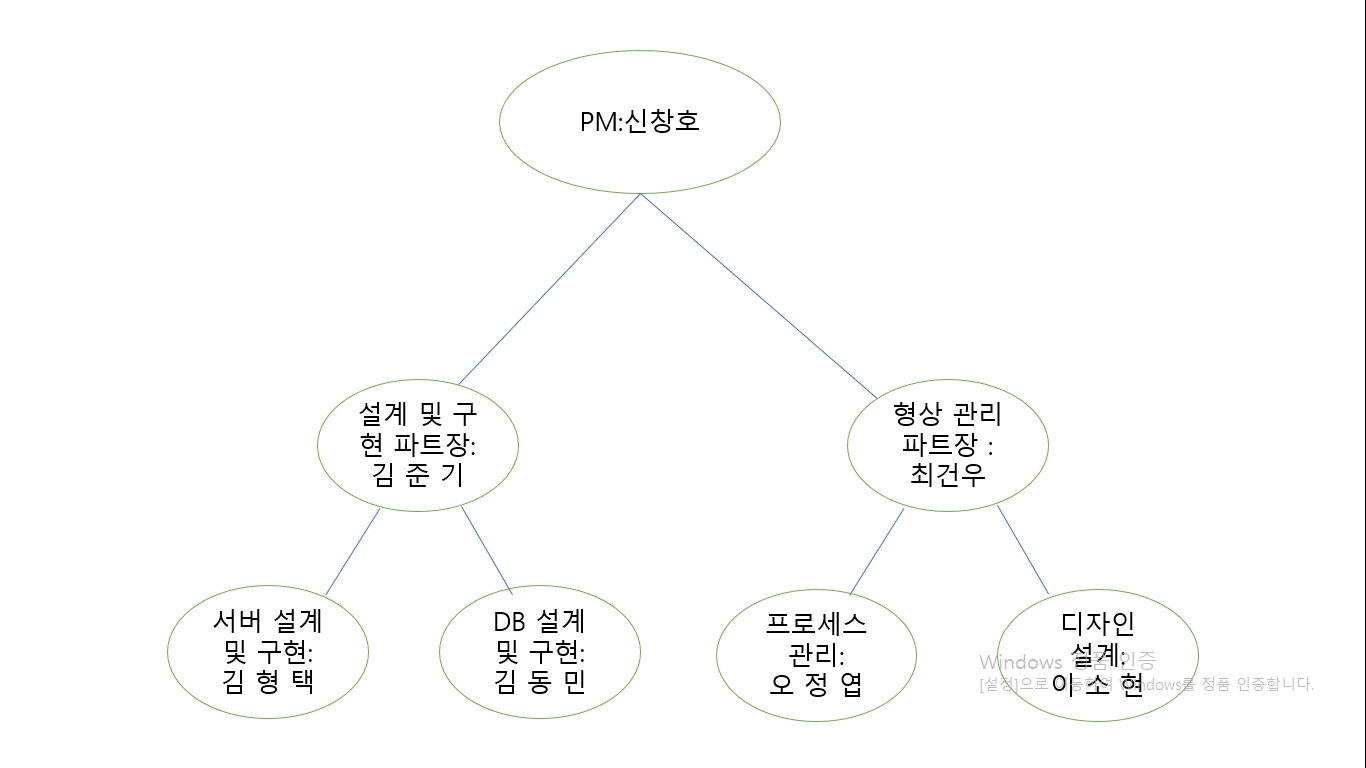
기존의 회사에서는 모바일 메신저가 성장함에 따라 카카오톡이나 페이스북 메신져, 라인 등을 활용하여 업무에 활용하였지만 이는 업무를 수행하는데 있어서 불편함으로 초래하였고 이러한 방식은 회사의 기밀 보안 유지에도 허술함을 보였다. 이러한 불편한 방식들을 해결하기 위해 야머(yammer), 슬랙(Slack), 아사나(ASANA) 등과 같은 협업관리 도구가 나타났다. 하지만 이러한 협업관리도구는 자료 통합, 기밀유출방지, 효율적인 의사소통 기능 등 다양한 기능이 존재하지만 실시간으로 여러 명이 자료를 관리하는 것은 쉽지 않다. 현재 클라우드가 발전함에 따라 ‘MS 오피스365’, ‘구글 문서’ 등 현재 실시간 공동편집 오피스 솔루션 이 등장하고 있는데 이러한 공동편집 솔루션을 사용하는 데 있어서 협업관리도구와 호환되지 않아 자료의 분산이 다시 문제점으로 발생하고 있다. 그래서 협업관리도구와 공동편집 오피스 솔루션 등을 융합하여 자료의 통합적 관리의 편의성을 목적으로하는 구글 API를 활용한 협업관리도구를 계획하였다.

* 1. **주요 일정**

|  |  |  |
| --- | --- | --- |
| **단계** | **일정** | **산출물 및 일정내용** |
| 프로젝트 계획서 | 2019.03.18~2019.04.01 | 프로젝트 계획서 |
| 요구사항 명세서 | 2019.04.01~2019.04.15 | 요구사항 명세서 |
| 설계 | 2019.04.15.~2019.04.27 | UI설계서 |
| 2019.04.27.~2019.05.13 | CLASS설계서 |
| 2019.04.27.~2019.05.06 | DB설계서 |
| 구현 및 개발 | N/A | N/A |
| 테스트 | 2019.05.13~2019.05.17 | 테스트 케이스 작성 |
| 2019.05.17~2019.05.21 | 테스트 진행 |
| 2019.05.21~2019.05.27 | 테스트 보고서 |
| 최종 보고서 작성 | 2019.05.27~2019.05.30 | 최종 보고서 |
| 최종 발표 | 2019.05.30~2019.06.03 | PPT준비 및 발표 리허설 |

* 1. **조직**

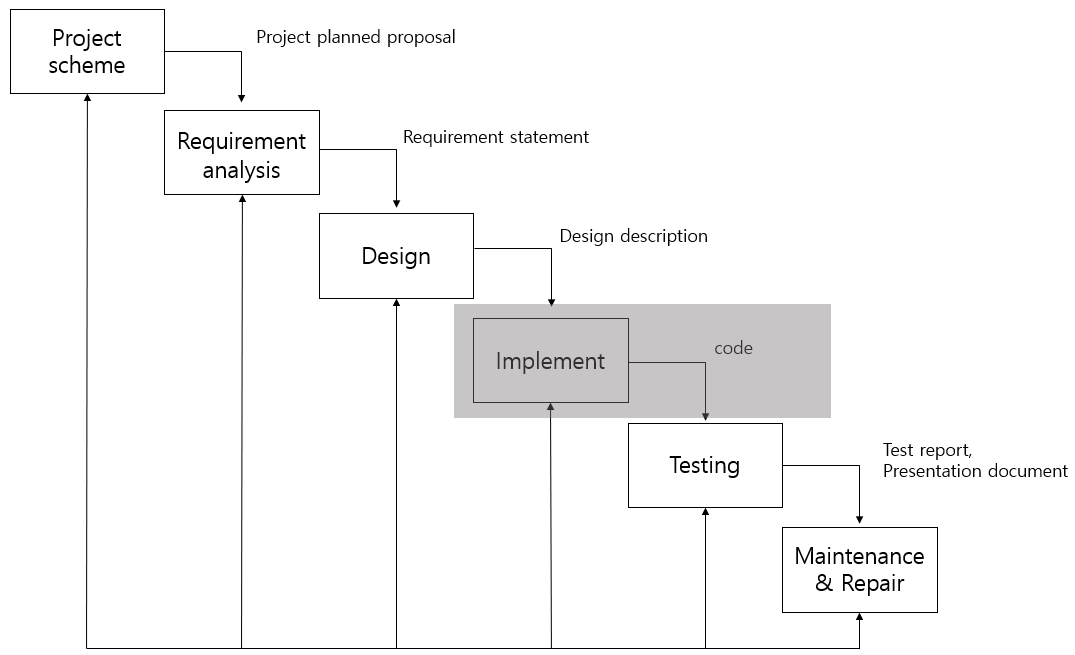
**1.3.1 조직도**



**1.3.2 역할 및 책임**

|  |  |  |
| --- | --- | --- |
| **팀원** | **역할** | **책임** |
| 신창호 | PM | 프로젝트 총괄 및 수행 |
| 김준기 | 모듈화 설계 | 프로그램의 프로젝트 모듈화 설계, 요구사항 분석 총괄 |
| 김형택 | 서버 설계 | 프로그램의 프로젝트 서버 설계, 프로젝트 계획서 총괄 |
| 김동민 | DB 설계 | 프로그램의 프로젝트 DB 설계 총괄 |
| 이소현 | 디자인 설계 | 프로그램의 프로젝트 UI, CLASS 설계 총괄 |
| 오정엽 | 프로세스 및 문서 관리 | 프로젝트의 프로세스 관리, 회의록 및 최종보고서 총괄 |
| 최건우 | 프로젝트 형상 관리 | 프로젝트의 형상 관리, 회의록 및 최종보고서 작성 |

* 1. **생명주기 모델 : Waterfall**



: 폭포수 모델(Waterfall Model)은 고전적인 생명주기 모델이다. 순차적으로 소프트웨어를 개발하는 모델로서 각 단계별로 정형화된 접근 방법으로 구체적인 문서화를 할 수 있고, 단계별로 산출물을 체크함으로써 프로젝트의 진행을 명확하게 할 수 있는 장점이 있다. 또한 고전적인 방법론으로서 적용 사례가 풍부하고 전체 과정에 대해 이해가 쉬우며 현재 단계에 대한 이해를 빠르게 할 수 있다. 하지만 폭포수 모델의 단점도 존재한다. 병행 작업이 힘들고 피드백에 대한 반복 단계 또는 고객 요구사항에 대한 상세한 반영이 어렵다. 위의 폭포수 모델을 프로젝트에 적용함으로써 발생할 수 있는 여러 상황을 경험하고 동시에 리스크 관리를 통해 프로젝트 설계에 관한 전반적인 프로세스를 이해하고 활용하려고 한다.

**1. 계획 수립**

- 타당성 검토 = 협업관리도구를 필요로 하는 고객의 요구 조건과 시스템 및 개발 환경을 고려하여 프로젝트 진행 여부를 판단, 개발 범위에 대한 사전 조사, 리스크 관리 계획

- 결과물: 프로젝트 계획서

**2. 요구사항 분석**

- 고객들이 기존 협업관리 도구를 사용할 때 느낀 불편사항과 도움이 되었던 기능을 조사

- UI에 대한 요구사항 조사

- 수집된 요구사항에 대한 사용자 요구사항 작성

- 결과물 : 요구사항 명세서, Usecase diagram

**3. 설계**

- 요구사항 명세를 준수하여 설계

- 실시간 협업관리도구에 포함되는 각각의 기능들간의 알고리즘을 작성

- 여러 기능들을 포함하는 전체적인 알고리즘을 작성

- 결과물 : 기본/ 상세 설계서(Class설계서, UI설계서,DB설계서), sequence diagram

**4. 테스트**

- 완성된 각각의 모듈을 정식으로 테스트하여 오류를 발견

- 통합테스트 = 여러명의 개발자가 개발한 소프트웨어가 결속하여 동작하는 지를 검사

- 시스템 테스트 = 전체 실시간 협업관리 도구의 시스템 동작을 검증

- 인수 테스트 = 사용자가 직접 협업관리도구를 사용하면서 고객들이 제시했던 요구사항에 대하여 검증을 함

- 결과물 : 테스트 계획서, 테스트 보고서

**5. 유지 보수**

- 실시간 협업관리도구를 사용하면서 나타나는 오류들을 수정

- 사용자의 요구에 따른 추가 기능을 개발

* 개발단계는 원래 프로젝트 단계에서라면 진행하나, 이번 프로젝트 특성상 개발부분은 생략
  1. **도구**

**1) Slack – 이슈 관리 도구**

웹 기반의 오픈소스 이슈(Issue) 관리 도구로서 새로운 기능, 결함, 할 일과 같이 주목해야하고, 해결해야 할 대상인 이슈들을 필요에 따라 이슈 목록을 관리하고 유지해주며 PM, 개발자, QA 등 역할에 따른 권한 제어, Gantt Chart 및 달력 등의 기능을 제공하며, Git과 같은 주요 형상 관리 도구와 같이 기존에 사용중이던 도구들의 알림을 통합, 연동을 지원해주는 도구이다. 또한 파일을 쉽게 공유하고, 그 파일에 대한 커뮤니케이션도 별도로 가능하며 API를 제공함으로써 개발자들이 원하는 정보를 슬랙을 통해 알림받을수 있도록 하는 도구이다.

**2) Google 스프레드 시트 - 문서 관리 도구**

웹 기반의 도구로써 관계자 외의 간섭을 안 받고 정해진 사람에게만 공유할 수도 있고 웹링크를 통해 제한없이 공유할 수도 있다. 자동저장 기능과 모든 수정내역이 업데이트 기록에 남고 필요하다면 이전 버전으로 복원도 가능하다. 협업과 자동화에 특화되어 있어 문서 작성 및 정리을 여러 사람이 협업하여 운영하여 시간절약에 용이하고, 공유자들이 실시간으로 문서 작업을 할 수 있으므로 항상 최신 버전으로 문서 작업을 할 수 있게 도와주는 문서관리 도구이다.

**3) Github – 형상 관리 도구**

오픈 소스 공개 프로젝트에 무료로 Git 저장소를 호스팅할 수 있는 특징을 가진 Github는 소스 코드와 같이 지속적으로 변경되는 파일들을 관리해주는 분산 버전 관리 시스템으로서 파일의 최신 버전 뿐만 아니라, 이전 버전까지의 전체 이력을 기록해주며 각 유저에게 Github의 저장소를 지원하여 개개인의 유저들이 Fork를 사용한 Clone기능, 커뮤니티의 활용 및 프로젝트 관리, 소스 변경 및 코멘트 등록, Commit, Push, Pull등의 기능들을 활용하여 서버와 동일한 환경에서 작업할 수 있도록 도와주는 도구이다.

**4) Microsoft PowerPoint, Word - 문서 작업 도구**

각종 문서들을 작성하기 위한 도구로써 효율적으로 작업할 수 있게 해주는 도구이다. 프로젝트 계획서, 요구사항 명세서, 설계서, 테스트 명세서, 최종 보고서 등에 이용할 것이다. 또한 다른 도구에 비해 가독성 있고, 깔끔한 특징이 있는 문서 작업 도구이다.

**5) Apache Maven – 빌드 도구**

프로젝트 객체 모델이라는 개념을 바탕으로 dependency와 라이브러리를 관리하고 표준화된 프로젝트 및 생명 주기 관리 기능을 제공하며, Maven은 Repository에서 필요한 모든 파일들을 직접 다운로드 할 필요 없이 해당 프로젝트로 불러와 주는 기능을 하는 플러그인을 기반으로 소스 코드로부터 배포 가능한 산출물을 만들어 내는 빌드 기능을 제공해주는 도구이다.

**6) MYSQL - 개발 도구**

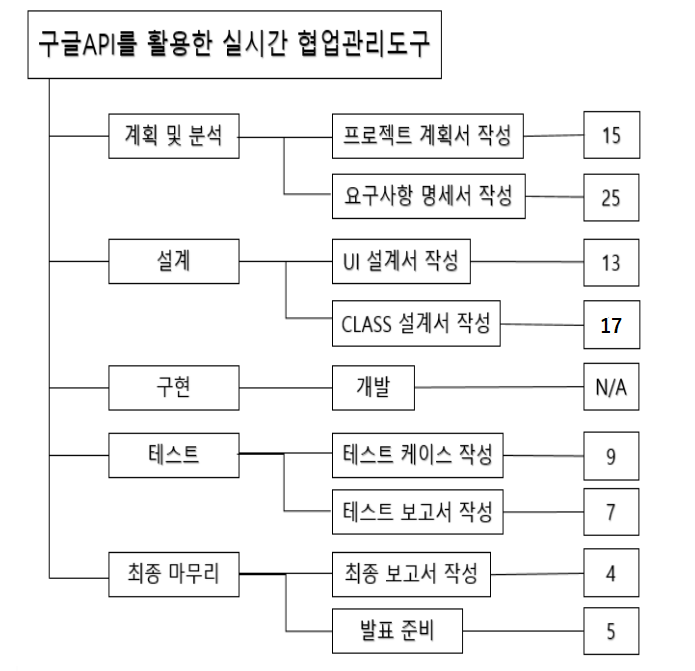
전세계적으로 널리 사용되고 있는 오픈소스 데이터베이스로써 표준 데이터베이스 질의언어 SQL을 사용하는 개방 소스의 관계형 DBMS(RDBMS)이고, 매우 빠르고 유연하며 사용하기 쉽다. 또한 다중 사용자, 다중 쓰레드를 지원하고 다양한 언어를 위한 응용프로그램 인터페이스(API)를 제공하며, 다양한 운영체제에서 사용할 수 있는 도구이다. 이번 프로젝트에서 DB설계시 이용할 예정이다.

**7) 카카오톡 - 커뮤니케이션 도구**

실시간 채팅 어플리케이션으로 팀원들간에 소통을 할 수 있게하는 도구이다. 문서 작업 시 게시판에 댓글을 달거나 어느정도 딜레이가 있는 다른도구들에 비해 빠르게 소통 가능하다는 장점이 있기때문에 회의 일정을 잡거나 문서 수정시 빠른대처가 가능한 도구이다.

1. **규모 산정**

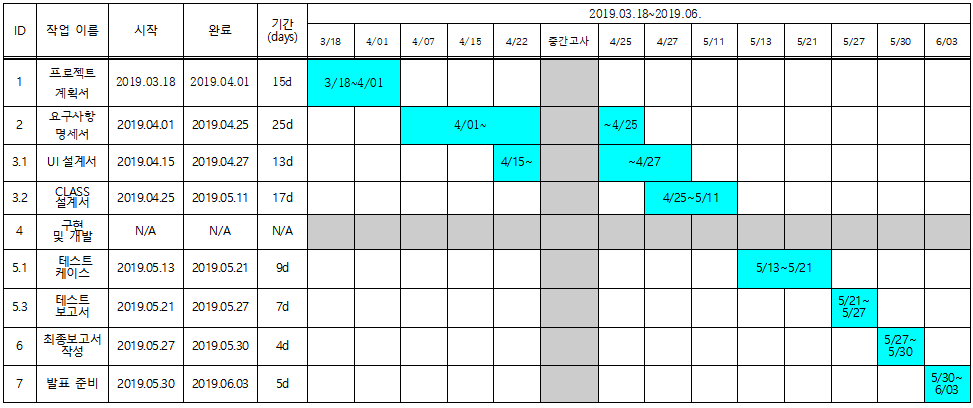
WBS(Work Breakdown Structure)



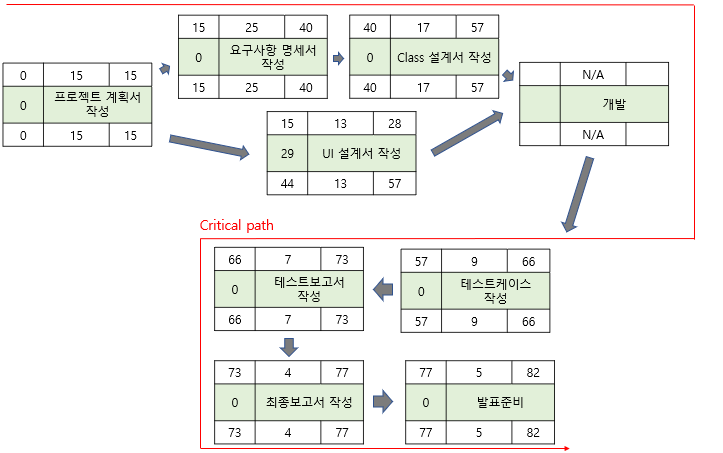
\* Day로 비용산정

1. **일정**

<Gantt Chart>



<Pert Chart>



1. **산출물 관리(팀원들 간의 관리 계획)**

프로젝트에서 대부분의 산출물은 프로젝트 계획서, 위험관리 계획서 및 위험관리 문서, 요구사항 관리 문서, 요구사항 관리문서, SW 설계문서, 테스트 계획서 및 결과서 등의 다양한 문서이다. 구글 드라이브를 이용해, 산출물들을 공유하고 실시간으로 다른 팀원들과 상의하며 산출물을 수정하는 방식을 이용한다. 이와 같은 방식을 이용하면 마구잡이로 산출물이 수정될 수 있다는 단점이 있기 때문에 일반적으로 문서의 버전은 파일명을 통하여 관리할 수 있도록 하고 추가적으로 PM이 산출물을 수정할 수 있는 권한과 시간을 부여하도록 한다. 예를 들면 “문서명(버전)\_yyyyMMdd\_작성자명.확장자”의 형태로 관리할 수 있도록 한다.

팀원들은 효율적인 산출물을 관리하기 위해서 프로젝트계획서나 요구사항 명세서 같은 문서들을 구글 드라이브를 통해서 작업을 하지만, 최종 산출물은 SVN, Github와 관리도구를 사용하여 버전 뿐만 아니라 변경이력까지 자동화하여 관리함으로써 산출물 관리에 편의성을 가져올 수 있도록 한다. 또한 소스코드와 같은 산출물들은 다른 API문서와 같은 산출물들과 같이 관리, 변경 및 수정함으로써 마구잡이식 개발을 억제할 수 있다. 그렇기 때문에 이를 이용하여 팀원들간의 효율적인 협업이 가능하게 한다.

위와 같은 방식들을 이용하여 코드 및 산출물을 관리할 수 있고 또한 코멘트 등록 및 메신저 기능을 활용하여 팀원들과 프로젝트의 전반적인 흐름을 공유할 수 있게 한다. 산출물들을 위와 같은 방식으로 관리한다면 팀원들간의 피드백을 손쉽게 주고 받을 수 있다는 장점이 있다. 또한 추후에 보수 및 유지 관리 시에 수정 또한 용이하게 할 수 있고 프로젝트와 관련하여 오픈소스 형태로 제공함으로써 다른 사람들에게도 좋은 피드백을 받아볼 수 있는 장점이 있다.

**5. 리스크 관리 계획**

|  |  |  |  |
| --- | --- | --- | --- |
| **리스크 요소** | **가능성** | **영향도** | **대처 방안** |
| 계획 수립 과정에서 많은 시간을 낭비할 수 있다.  ex) 팀원 간의 사정으로 오프라인 미팅이 원활하지 않을 수 있다. | 상 | 상 | 개발 초기 계획 단계에서 고객의 요구 사항에 기반하여 팀원들간의 충분한 협의와 토의를 통해 발생할 수 있는 문제점을 줄여나감으로써 시간을 단축한다. |
| 요구 사항 변경에 유연하게 대처할 수 없다.  ex) 설계 및 문서화 진행 후 에는 수정이 힘들다. | 중 | 중 | 개발 초기 계획 단계에서 고객의 요구사항이 지속적으로 변화하고 요구될 수 있다는 점을 감안하고 프로젝트 진행에 있어서 피드백이 이뤄질 수도 있다는 점을 염두한다. |
| 각 단계를 문서화해서 진행하다보면 문서 작성 시간과 다른 업무에 대해 부담이 될 수 있다.  ex) 프로젝트 수행 기간 중 절차에 따라 문서작성이 많은 시간을 소요할 수 있다.. | 상 | 상 | 문서 작성을 확실히 하면 반대로 장점으로 작용할 수 있다. 하지만 문서 작성에 많은 시간이 소요되는 것은 문제가 발생할 수 있으므로 문서 작성을 체계적으로 하여 시간을 줄일 수 있도록 한다. |
| 테스트 단계에서 심각한 문제가 발생하면 전체적인 프로젝트에 차질이 생길 수 있다.  ex) 각 팀원들이 경험이 부족하여 설계적으로 오류가 발생했다. | 상 | 상 | 프로젝트 진행 과정의 각 단계에서 중요도 높은 단위에 대해 단위 테스트를 실시하여 최종 테스트시 발생할 수 있는 문제점을 줄일 수 있도록 한다. |
| 절차가 복잡해짐에 따라 실제 프로젝트를 진행하는 데 있어 산출물의 퀄리티가 떨어질 수 있다.  ex) 문서작성 및 절차 진행에 따라 개발하는 기간이 줄어들 수 있다. | 중 | 중 | 각 역할 업무의 분담을 조금 간소화하고 부담을 줄임으로써 전체적인 프로젝트 진행에 있어 산출물의 퀄리티를 향상시킬 수 있는 방향으로 진행하도록 한다. |
| 협업이 제대로 이루어지지 않을 시에 전체적인 프로젝트 진행기간과 비용을 증가시킬 수 있다.  ex) 팀원 간의 의견 마찰, 연락이 잘 안되는 경우 | 중 | 상 | 팀원 간의 충분한 대화와 시간을 통해 생길 수 있는 갈등이나 문제점 등을 해결할 수 있도록 노력하고 프로젝트를 진행하면서 의견 수립에 있어 구체적인 절차와 과정을 제시하도록 한다. |