**산출물 관리**

모든 산출물은 Github에 올려서 모든 팀원들과 공유한다.

Redmine을 통해 일감을 관리하고 산출물이 효과적으로 관리 될 수 있게 한다. .

변경 이력은 문서 수정한 사람이 작성하며 정확하고 간결하게 명시한다.

문서 버전은 V1.0.0 으로 시작한다.

* Github
* 프로젝트의 모든 산출물은 Github의 repository(18-1\_7\_SHORTSTORMERS)

에 올림

* doc 폴더

회의록, 요구사항 명세서, 프로젝트 계획서, 설계서, 테스트 결과 보고서 등 모든 문서를 저장

* src 폴더

프로젝트에 사용된 모든 소스코드를 저장

* 파일명
  + 월\_날짜\_팀 명\_회의록.doc
  + 문서 이름\_팀 명.doc
* 클래스 명.java
* Redmine
  + 일감 유형에 따라 산출물 관리

Ex)일감 유형: 계획 -프로젝트 계획서

분석 –요구사항 명세서

설계 –Class 설계서, UI 설계서

구현 -Source Code

테스팅 -테스트 결과 보고서

* 버전 관리
  + 변경사항이 생기는 경우 규칙에 따라 commit message를 작성하고, commit 하여 버전을 관리
  + V1.0.0 (첫 버전)

**리스크 관리 계획**

|  |  |  |  |
| --- | --- | --- | --- |
| 리스크 요소 | 가능성 | 영향도 | 대처 방안 |
| 요구사항 변경 | 1 | 상 | -요구사항 변화에 따라 빠르게  문서와 소스코드를 추가 및 수정 한다.  -소스코드 개발 초기부터 추가 및 수정 요구사항에 대해 적응력이 높도록 구현하도록 노력한다. |
| 팀원들 간의  의사소통 문제 | 3 | 중 | -PM이 주도하여 합리적인  의사소통을 통해 팀원간의 합의를 이끌어 내고 합리적으로 역할분담을 한다. |
| 개발 기간 부족 | 3 | 중 | -일정계획에서 기간이 오래 걸릴것 같은 부분에 여유 기간을 두고, 일정관리를 철저하게 한다. |
| 개발 능력 부족 | 3 | 상 | -메인 개발자가 개발을 진행 하 되 부족한 부분은 팀원들이 모두 나서서 보충한다.  -구현 가능한 수준까지 최대한 개발 하다가 미흡 부분에 대해서는 시간과 비용을 다시 설계한다. |
| 산출물 소실 | 5 | 최하 | -Github repository에 철저한 데이터 원본 및 버전 관리를 한다.  -팀원들 각자 산출물을 백업 해 논다. |
| 에러발생 | 2 | 하 | -일정에 버퍼 기간을 두어서 에러가 발생하더라도 팀원들과 협력하여 에러를 해결한다.  - |

**최상:** 프로젝트 진행에 어려움을 주고 프로젝트가 실패할 수 있다.

**상:** 프로젝트 진행에 어려움을 준다.

**중:** 일정에 지연이 생긴다.

**하:** 프로젝트 진행에 어려움을 줄 수 있지만 쉽게 해결할 수 있다.

**최하:** 문제 없다.

**가능성:** 1, 2, 3, 4, 5 (1이 발생 가능성이 가장 높음)