

프로젝트 명세서

소프트웨어 유지보수

목차

1. 과제 개요	3
2. 과제 상세	4
3. 산출물 제출.....	4
4. 추천 학습 방법.....	5

1. 과제 개요

본 과제는 프로젝트의 여러 단계 중 유지보수에 대해서 이해하고 해당단계에서 필요한 과제를 수행해보는 과정입니다

소프트웨어 유지보수란 제품으로 개발된 후 User 가 사용하기 시작하면서부터 폐기될 때까지 오류를 수정하거나 새로운 기능을 추가하기 위해 이를 변경하는 과정을 의미합니다

유지보수의 유형에는 Corrective Maintenance(수정 보수), Adaptive Maintenance(적응 보수), Perfective Maintenance(향상 보수), Preventive Maintenance(예방 보수)등이 있습니다

유지보수

소프트웨어를 인수, 설치한 후에 발생하는 모든 공학적 작업

원인에 따라 수리(Corrective) 보수, 적응(Adaptive) 보수, 완전화(Perfective) 보수, 예방(Preventive) 보수 등이 있습니다.

유지보수를 쉽게 하려면 시험 용이성, 이해성, 수정 용이성, 이식성을 고려해야 합니다.

수리보수(Corrective): 검사에서 밝혀지지 않은 모든 잠재적인 오류를 수정하기 위한 보수 형태로 수정과 진단을 포함

적응보수(Adaptive): 소프트웨어 수명 기간 중에 발생하는 하드웨어, 운영체제 등 환경의 변화에 반영하기 위해 수행

완전화보수(Perfective): 현재 수행 중인 기능의 수정, 새로운 기능 추가, 전반적인 기능 개선 등의 요구를 수행, 제일 많은 비용

예방보수(Preventive): 장래의 유지보수성 또는 신뢰성을 개선하거나 소프트웨어 오류발생에 대비하여 미래 예방 수단을 강구

유지보수 비용 산정 공식

$$M = P + Ke^{(c-d)}$$

M:유지 보수를 위한 노력(인원/월)
P: 생산적인 활동에 드는 비용
K: 통계값에서 구한 상수
c: 복잡도
d: 소프트웨어에 대한 지식의 정도

본 과제를 통해서, 유지보수 단계를 경험해보고 해당 단계에서 체득할 수 있는 게 어떤 것인지 이해하시길 바랍니다.

2. 과제 상세

Preventive Maintenance(예방 보수)는 장애에 유지보수성 또는 신뢰성을 보장하기

위해 선제적으로 진행하는 행위를 의미합니다

1 학기에 본인이 진행한 관통프로젝트 중에서 유지보수의 Preventive Maintenance(예방 보수) 유형의 한가지인 Code Refactoring 을 진행합니다

-본인이 작성한 소스에서 가독성이 낮은 코드나 동일 로직이 중복 적용된 부분을

찾아내어 확장성 있게 모듈화하는 등의 Refactoring 작업을 진행합니다

3. 산출물 제출

- 제출 위치

<https://lab.ssafy.com/s10-study/self-project> 의 산출물 제출가이드.docx 참조

- 작성 내용

산출물 제출	생성한 프로젝트를 gitlab 에 제출한다. README.md 에 학습 내용과 “리팩토링 전/후 소스” 를 포함한다.
-----------	---

4. 추천 학습 방법

아래 경로의 문서를 통해 소프트웨어 유지보수에 대한 이해도를 높여보시기 바랍니다

<http://contents.kocw.or.kr/KOCW/document/2014/deagucatholic/kimhangkon1/13.pdf>

또한 Preventive Maintenance(예방 보수) 유형의 한가지인 Code Refactoring에 대한

상세 학습을 위해 <https://refactoring.guru/>를 참고하시기 바랍니다