

# 마감 하루 전

## Category Partitioning Testing Tool

**Project Team**

**T1**

**Date**

**2017-04-13**

---

**Team Information**

201111334 김강욱

201211339 김진욱

201312243 김동권

201510411 이소영

# INDEX

**Activity 1001. Define Draft Plan**

**Activity 1002. Create Preliminary Investigation Report**

**Activity 1003. Define Requirements**

**Activity 1004. Record Terms in Glossary**

**Activity 1005. Implement Prototype**

**Activity 1006. Define Business Use Case**

**Activity 1007. Define Business Concept Model**

**Activity 1008. Define Draft System Architecture**

**Activity 1009. Define System Test Case**

**Activity 1010. Refine Plan**

## Activity 1001. Define Draft Plan

### 1. Motivation

점차 복잡하고 다양한 프로그램들이 개발됨에 따라 프로그램의 Spec 범위도 넓어지고 있다. '한정된 시간에서 꼭 필요한 Test 를 수행함으로써 가능한 적은 시간 내에 오류 없는 프로그램을 만드는 것' 에 대한 수요가 증가하고 있다.

이러한 추세에서 자연어로 구성 된 Specification 문서는 테스터와 기획자 간의 간격을 더 넓히며 시간의 효율성을 떨어트린다. Category Partitioning은 Spec 명세를 독립적으로 테스트 할 수 있는 기능 단위로 분해하고 조건 별로 Category를 설정함으로써 테스터에게 효율적인 Test case 조합을 제공한다. 본 Testing Tool이 제공하는 효율성은 기업뿐 만 아니라 학교에서 진행되는 프로젝트에도 동일하게 적용되어야 하며 이에 따라 Category Partitioning Testing Tool 을 개발하게 되었다.

### 2. Project Objectives

사용자가 정한 Category 별 Constraint에 대해 Ideal한 Test case를 조합하고, 이 결과를 사용자가 정한 우선순위에 따라 중요한 Test case에 대한 정보도 제공하는 것을 목적으로 한다.

중요하다고 생각되는 Value들에 대해 가중치를 줄 수 있고 Test case 산출 시 이 가중치에 대한 계산 값을 함께 제공한다.

- 사용자가 입력 한 Category에 대해 Test case를 조합한다.
- 팀원들이 각자 생각하는 중요한 Test case 조합에 대해 명시한다.
- Admin 권한을 가진 팀장이 프로그램 내에서 팀원들에게 전달 받은 Test case 조합에 대해 Constraints와 가중치에 대해 한번 더 검토한다.
- Test Case 와 함께 먼저 Test가 수행되어야 하는 Case에 대한 정보를 제공받는다.

### 3. Project Scope

'마감 하루 전'은 중요한 Test case를 먼저 Test 할 수 있는 계산 된 결과 값을 제공함으로써 사용자로 하여금 효율적인 Testing이 가능하도록 한다. 프로그램 별로 중요하거나 오류 가능성이 많은 Case가 존재하고, 한정된 시간 내에 오류 수정 및 프로그램의 효율적인 개발을 위해 이러한 Test가 먼저 수행되어야 한다는 점에 초점을 맞추었다.

### 4. Functional requirements

- Open File
- Parsing File
- Apply to the Weight of the value
- Modify Weight
- Input Essential Test Case
- Compare with Test Case
- Modify Constraint
- Make Test Case
- Arrange by Weight Average
- Make Final Test Case File

### 6. Non-Functional requirements

- 5초 이내로 결과값이 나와야 한다.
- 가중치와 누락된 Test case를 파악하기 쉬워야 한다.

### 7. Resource Estimation

- 1) Human Efforts (Man - Month): 4 - 3
- 2) Human Resource: 컴퓨터공학과 2명, 다전공 2명

3) Project Duration: 16주

4) Budget

	부분 계	비고
인건비	360	
활동비	12	
계	372	

**Activity 1002. Create Preliminary Investigation Report.**

**1. Alternative Solutions**

- 1) 기존에 있는 Category Partitioning Testing Tool을 사용한다.
- 2) 기존에 있는 Category Partitioning Testing Tool을 산다.
- 3) 타 소프트웨어 개발업체에 외주를 의뢰한다.

**2. Project Justification (Business Needs)**

1) Cost

- ① 외주업체에 비해 개발 비용이 저렴하다.

2) Duration

- ① 2017년 1학기 안에 완성해야 한다.

3) Risk

Plan Management, OOPT Skill, Programming Skill, UML Skill, 시험

4) Effect

- ① 유지보수의 이점이 있다.
- ② 빠른 시간 안에 Test Case를 조합할 수 있다.
- ③ 프로그램을 독점적으로 이용할 수 있다.

**3. Risk Management**

Risk	Probability	Significance	Weight
Plan Management	5	5	25
OOPT Skill	4	10	40
Programing Skill	3	8	24
UML Skill	5	9	45
시험	10	10	100

**4. Risk Reduction Plan**

Risk	Reduction Plan
Plan Management	관련 사이트와 서적을 활용하여 도움을 얻는다.
OOPT Skill	수업강의 자료와 교수님, 조교님의 도움을 얻는다.
Programming Skill	Programing 관련 서적을 읽고 선배들의 도움을 얻는다.
UML Skill	UML관련 서적을 읽고 선배들의 도움을 얻는다.
시험	조원들끼리 서로 협력하여 서로 도움을 주고 받는다.

**5. Market Analysis**

1) 기존의 Category Partitioning Test Tool이 많지 않다. 또한 자연어로 명시된 Programming Specification에 대한 요구가 점점 증가할 것이다. 이에 따라 한정된 시간을 활용하기 위해 우선적으로 검증해야 하는 Test Case가 필요할 것이며, 시간내에 모든 오류를 잡고 출시해야 한다.

**6. Other Managerial Issues**

1) 2017년 6월까지 개발을 완성하여 A+을 받아야 한다.

**Activity 1003. Define Requirements**

**1. Interview**

- 'grepp' Software Engineer와 인터뷰

Q1) 현업에서 쓰이는 Test case 조합이 있는가?

A1) TDD(Test Driven Development)를 사용

1. 내가 얻고자 하는 결과물 생각(기능) 및 어떻게 테스트 할 것인가?
2. 간단한 테스트를 작성(기능에 최적화 된)
3. 테스트를 또 작성하는데, 이번에는 Fail이 뜨도록 일부러 작성
4. 마지막 테스트는 Pass가 나오도록 작성
5. 결과를 보며, 중복 코드 혹은 Logic, 불분명한 코드, 리팩토링 가능한 코드를 찾아서 고친다.
6. 마지막 모든 테스트 돌려서 Pass하도록 정리.
7. 완료 되었다고 생각할 때까지 테스트 => 깃에 올려서(다른 사람 코드와 합침) => 다시 테스트 케이스 돌림 => Deploy(서버에 올림) => 테스트 => Release

Q2) 제품 출시 전 얼마나 많은 Test case를 테스트하는가?

A2) 1. 제품에 따라 다르다. 유저에게 Critical Issue가 생기는 코드는 Function 하나 하나 그에 대한 Test case를 만드는 방식  
 2. 'Codly'와 같은 예쁘게 보이고, Contents가 중요한 경우, 테스트 케이스를 따로 만들지는 않고 있다. 대신 스스로 예측할 수 있는 모든 범위의 에러를 최대한 방지 하려 노력.  
 테스트를 완벽히 하지 않아서 우리는 브라우저 Compatibility 가 조금 문제가 있음

Q3) 한정된 시간에서 Test case를 실제 모두 수행하는 것이 부담되지 않는가?

A3) 1. 부담이 된다.  
 2. 하지만, 대표님 방침으로 테스트 케이스는 엄청 중요한 거라서 무조건 하고, 서로 코드리뷰 시간도 갖는다.  
 3. 서로의 코드가 규칙을 어길 경우 그에 대한 조언 및 리뷰를 확실히 해준다. (때론 갈등이 되기도 한다.)

Q4) 어떻게 하면 효율적인 Test가 가능할까?

A4) 습관을 들이는 수 밖에, 코드에 집착하지 말 것, 고집 피우지 말 것, 받아 들일 것.

Test case 작성에서 가장 큰 이슈는, 사람간의 관계라고 생각한다.

얼만큼 테스트 케이스가 중요하나를 대하는 사람들의 자세 및 규칙을 따르려는 의지.

**2. Functional Requirements**

Function	Description
Open File	엑셀 파일 열기
Parsing File	엑셀 파일을 읽어서 프로그램 내부에 저장
Apply to the Weight of the value	Value에 사용자가 입력한 가중치를 적용
Modify Weight	가중치 수정
Input Essential Test Case	중요 테스트 케이스 입력
Compare with Test Case	프로그램에 의해 조합 된 테스트 케이스와 사용자가 입력한 중요 테스트 케이스를 비교
Modify Constraints	제한 조건 수정
Make Test Case	테스트 케이스 생성
Arrange by Weight Average	가중치 평균에 따른 정렬
Make Final Test Case File	최종 테스트 케이스 파일 생성

Ref.#	Function	Category
R 1.1	Open File	Evident
R 1.2	Parsing File	Hidden
R 2.1	Modify Weight	Evident
R 2.2	Apply to the Weight of the value	Hidden
R 3.1	Input Essential Test Case	Evident
R 3.2	Compare with Test Case	Hidden
R 4.1	Modify Constraints	Evident
R 5.1	Make Test Case	Evident
R 5.2	Arrange by Weight Average	Hidden
R 6.1	Make Final Test Case File	Evident

**3. Operating Environments**

- 1) OS: Windows 7, Windows 10
- 2) 개발언어: Java

**4. Development Environments**

- 1) OS: Windows 7, Windows 10
- 2) IDE: Eclipse
- 3) 개발 언어: Java

### Activity 1004. Record Terms in Glossary

Term	Description	Remarks
Open	열다	
File	파일	
Parsing	파일을 읽어서 프로그램 내부에 저장	
Input	입력한다	
Weight	가중치 값	
Assign	할당한다	
Value	가중치 설정 후 계산된 값	
Modify	수정한다	
Essential Test Case	중요 Test case	
Show	보여준다	
Check	체크한다	
Constraints	Single, Error, Property 등의 제한 조건	
Compare	비교한다	
Test Case	Representative Value 조합으로 만들어진 Test case	
Make	생성한다	
Arrange	정렬	
Average	평균	

### Activity 1005. Implement Prototype

The screenshot shows a software interface with a table titled 'Category1.xlsx' and a 'Generate' button. The table contains test case data with columns: Index, Weight, Value1, Value1 Weight, Value1 Constraints, and Value2. The interface is in Korean.

Index	Weight	Value1	Value1 Weight	Value1 Constraints	Value2
102	1	소스파일 매우 많이 추가	1	Single	
104	1	소스 파일이 아닌 파일 추가	1	Error	
110	1	아무것도 선택하지 않고 파일 삭제	1	Single	
100.200.300.310.320.330.340.350.400.410.42	0.7	소스 파일 1개 추가	0.5	property AD1	클론제강 버튼 클릭
101.111.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		파일을 선택한 후 1개 삭제
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		

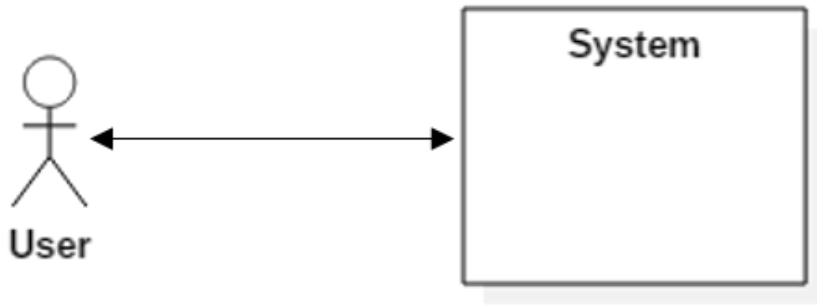
Essential Test Case

Index	Weight	Value1	Value1 Weight	Value1 Constraints	Value2
100.200.300.310.320.330.340.350.400.410.420	0.7	소스 파일 1개 추가	0.5	property AD1	
102.111.200.300.310.320.330.340.350.400.410.420	0.6	소스파일 매우 많이 추가	0.7	Single	파일을 선택한 후 1개 삭제
100.112.200.300.310.320.330.340.350.400.410.420	0.6	소스 파일 1개 추가	0.7	property AD1	파일을 삭제하지 않는다.

Input Essential Test Case

### Activity 1006. Define Business Use Case

## 1. Define Business Use Case



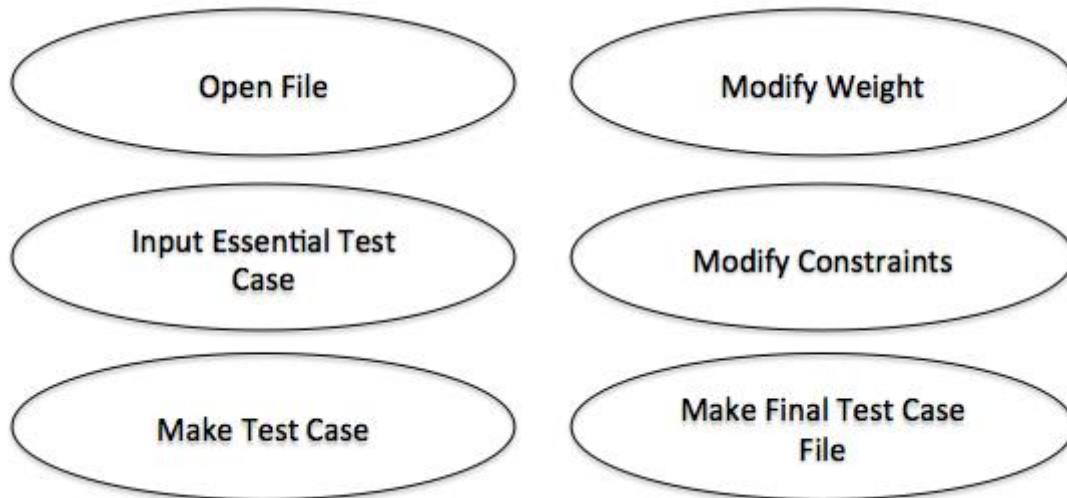
## 2. Identify and Describe Actors

User: Test Case 목록 생성을 위해 시스템과 상호작용하는 Actor

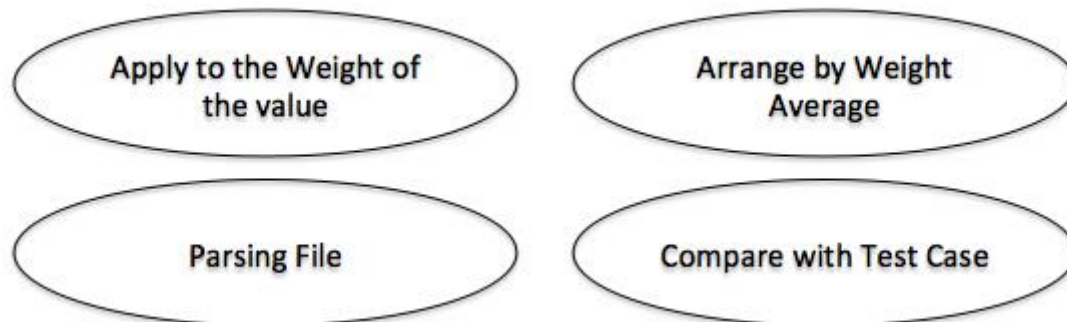


### 3. Identify Use-Case

#### 1) Actor based



#### 2) Event based



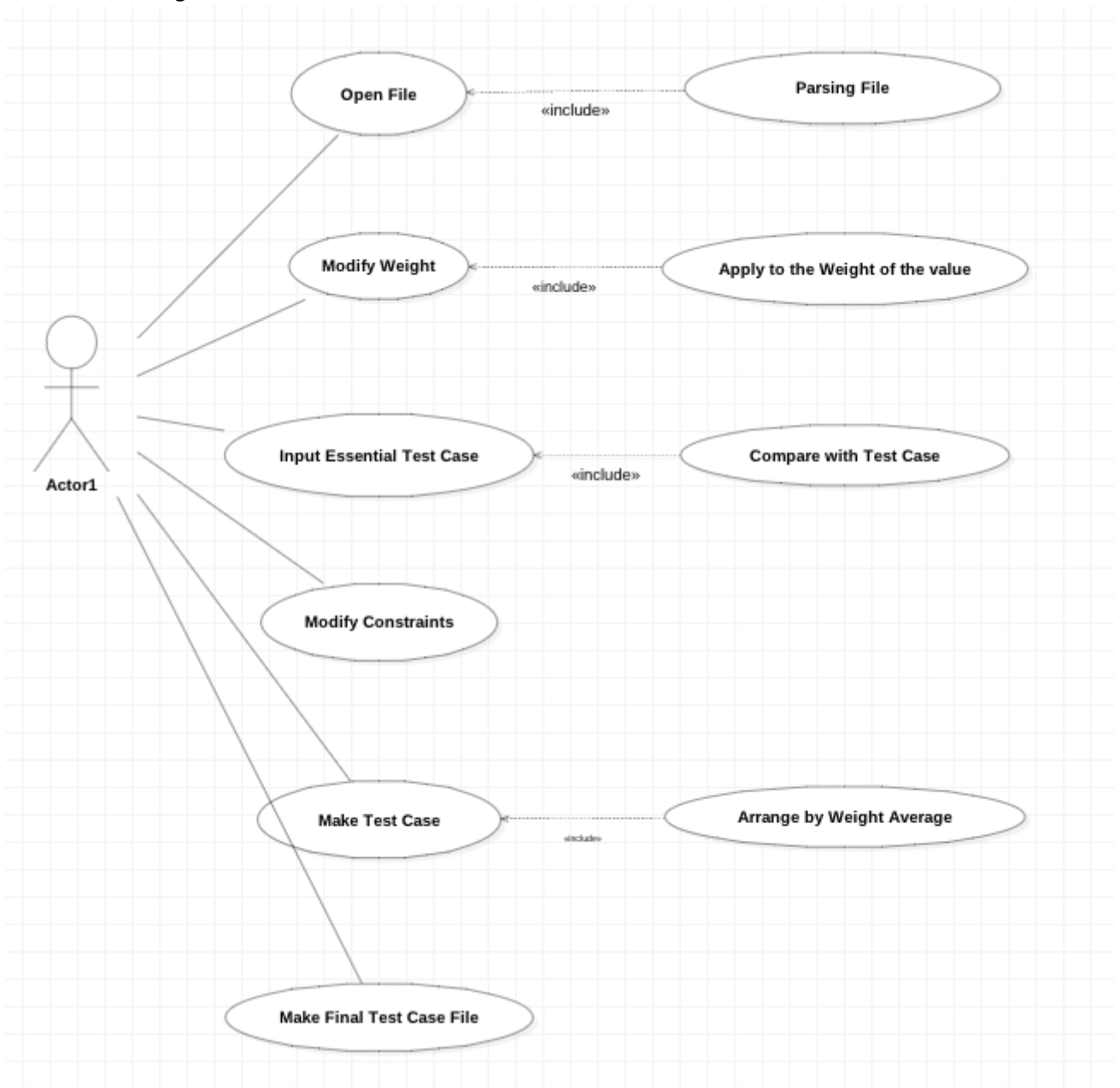
**4. Allocate system functions into Related Use-Case**

Ref.#	Function	Use-Case
R 1.1	Open File	Open File
R 1.2	Parsing File	Parsing File
R 2.1	Modify Weight	Modify Weight
R 2.2	Apply to the Weight of the value	Apply to the Weight of the value
R 3.1	Input Essential Test Case	Input Essential Test Case
R 3.2	Compare with Test Case	Compare with Test Case
R 4.1	Modify Constraints	Modify Constraints
R 5.1	Make Test Case	Make Test Case
R 5.2	Arrange by Weight Average	Arrange by Weight Average
R 6.1	Make Final Test Case File	Make Final Test Case File

**5. Category Use-Case**

Function	Use-Case
Open File	Primary
Parsing File	Primary
Modify Weight	Primary
Apply to the Weight of the value	Primary
Input Essential Test Case	Primary
Compare with Test Case	Primary
Modify Constraints	Primary
Make Test Case	Primary
Arrange by Weight Average	Primary
Make Final Test Case File	Primary

6. Draw a Use-Case diagram



7. Describe Use-Case

Use Case	1.Open File
Actors	User
Description	엑셀 파일을 불러온다.

Use Case	2. Parsing File
Actors	System
Description	엑셀 파일을 읽어와서 프로그램 내부에 저장한다.

Use Case	3. Modify Weight
Actors	User

Description	Value에 할당 된 가중치를 수정한다.
-------------	------------------------

Use Case	4. Apply to the Weight of the value
Actors	System
Description	사용자가 수정한 가중치를 Value에 할당하고, Test Case 가중치를 수정한다.

Use Case	5. Input Essential Test Case
Actors	User
Description	중요 테스트 케이스를 입력한다.

Use Case	6. Compare with Test Case
Actors	System
Description	중요테스트케이스와 테스트 케이스를 비교한다.

Use Case	7. Modify Constraints
Actors	User
Description	제한 조건을 수정한다.

Use Case	8. Make Test Case
Actors	System
Description	테스트 케이스를 생성한다.

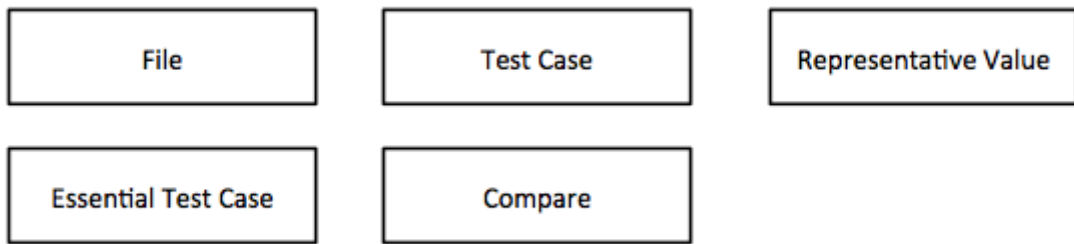
Use Case	9. Arrange by Weight Average
Actors	System
Description	테스트 케이스를 구성하는 Value들의 가중치 평균값을 구하여 정렬한다.

Use Case	10. Make Final Test Case File
Actors	User
Description	테스트 케이스 파일을 문서화 하여 생성한다.

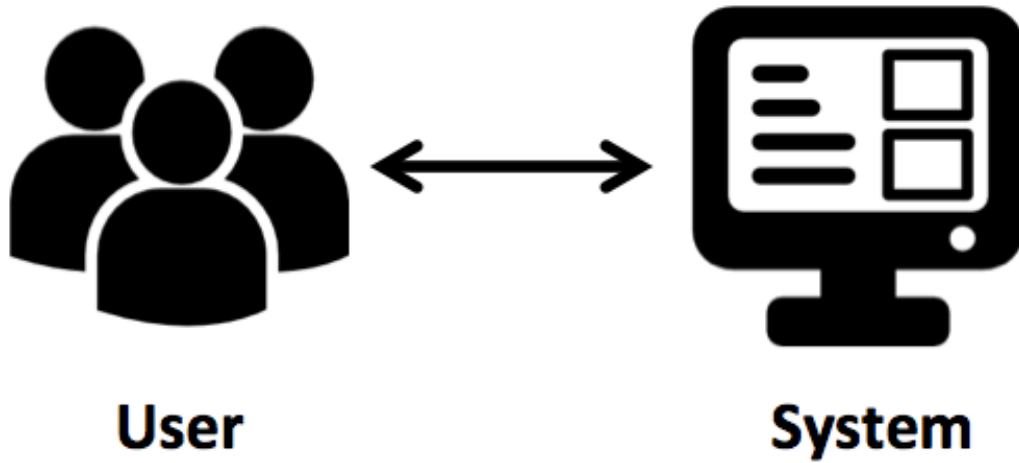
## 8. Rank Use-Case

Use-Case Number & Name	Rank
1. Open File	High
2. Parsing File	High
3. Modify Weight	High
4. Apply to the Weight of the value	High
5. Input Essential Test Case	High
6. Compare with Test Case	High
7. Modify Constraints	High
8. Make Test Case	High
9. Arrange by Weight Average	High
10. Make Final Test Case File	High

**Activity 1007. Define Business Concept Model**



**Activity 1008. Define Draft System Architecture**



**Activity 1009. Define System Test Case**

Test Number	Test 항목	Description	Use Case	System Function
1	파일 열기 시험	파일 열기 버튼을 눌러 엑셀 파일을 여는 기능 Test	Open File	R 1.1
2	파일 파싱 시험	엑셀 파일을 읽어서 프로그램에 넣어주는 기능 Test	Parsing File	R 1.2
4	가중치 수정 시험	가중치를 수정하는 기능 Test	Modify Weight	R 2.1
3	가중치 적용 시험	사용자가 입력한 가중치를 적용하는 기능 Test	Apply to the Weight of the value	R 2.2
5	중요 테스트 케이스 입력 시험	입력 버튼을 눌러 필수 테스트 케이스를 입력하는 기능 Test	Input Essential Test Case	R 3.1
6	테스트 케이스 비교 시험	프로그램에 의해 조합된 테스트 케이스와 사용자가 입력한 중요 테스트 케이스를 비교하는 기능 Test	Compare with Test Case	R 3.2
7	제한 조건 수정 시험	제한 조건을 수정하는 기능 Test	Modify Constraint	R 4.1
8	테스트 케이스 목록 생성 시험	테스트 케이스 목록 생성 기능 Test	Make Test Case	R 5.1
9	가중치에 의한 정렬 시험	가중치에 의한 정렬 Test	Arrange by Weight	R 5.2
10-1	최종 테스트 케이스 목록 파일 생성 시험	최종적으로 생성된 테스트 케이스를 파일 출력 버튼을 눌러 파일로 내보내는 기능 Test	Make final Test Case File	R 6.1
10-2	테스트 케이스 목록을 생성하는데 걸리는 시간 시험.	Test Case가 조합되는 시간이 5초 미만인지 Test	Make final Test Case File -> Nonfunctional Requirements	

**Activity 1010. Refine Plan****1. Project Scope**

'마감 하루 전'은 중요한 Test case를 먼저 Test 할 수 있는 계산된 결과 값을 제공함으로써 사용자로 하여금 효율적인 Testing 이 가능하도록 한다. 프로그램 별로 중요하거나 오류 가능성이 많은 Case가 존재하고, 한정된 시간 내에 오류 수정 및 프로그램의 효율적 개발을 위해 이런 Test가 먼저 수행되어야 한다는 점에 초점을 맞추었다.

**2. Project Objectives**

사용자가 정한 Category 별 Constraint에 대해 Ideal한 Test case를 조합하고, 이 결과를 사용자가 정한 우선순위에 따라 중요한 Test case에 대한 정보도 제공하는 것을 목적으로 한다. 중요하다고 생각되는 Value 들에 대해 가중치를 줄 수 있고 Test case 산출 시 이 가중치에 대한 계산 값을 함께 제공한다.

- 사용자가 입력한 Category에 대해 Test case를 조합한다.
- 팀원들이 각자 생각하는 중요한 Test case 조합에 대해 명시한다.
- Admin 권한을 가진 팀장이 프로그램 내에서 팀원들에게 전달 받은 Test case 조합에 대해 Constraints와 가중치에 대해 한번 더 검토한다.

[마감 하루 전]

T1

14

- Test Case 와 함께 먼저 Test가 수행되어야 하는 Case에 대한 정보를 제공받는다.

**3. Functional Requirements**

- Open File
- Parsing File
- Apply to the Weight of the value
- Modify Weight
- Input Essential Test Case
- Compare with Test Case
- Modify Constraint
- Make Test Case
- Arrange by Weight Average
- Make Final Test Case File

**4. Performance Requirements**

가중치에 따른 정렬한 결과가 5초 이내로 나와야 한다

**5. Operating Environment**

- 1) OS: Windows 7, Windows 10
- 2) IDE: Eclipse
- 3) 개발 언어: Java

**6. Resources**

- 1) Human Efforts (Man – Month): 4 – 3
- 2) Human Resource: 컴퓨터공학과 2명, 다전공 2명
- 3) Project Duration: 16주
- 4) Budget

	부분 계	비고
인건비	360	
활동비	12	
계	372	

7. Scheduling

Stage	Phase(00X0)/Activity(000X)	Schedule(Week)									
		1	2	3	4	5	6	7	8	9	10
1000. Plan & Elaboration	1001. Define Draft Plan	█									
	1002. Create Preliminary Investigation Report	█									
	1003. Define Requirements	█									
	1004. Record Terms in Glossary	█									
	1005. Implement Prototype	█	█								
	1006. Define Business Use Case		█								
	1007. Define Business Concept Model		█								
	1008. Define Draft System Architecture		█								
	1009. Define System Test Case		█								
	1010. Refine Plan		█	█							
2000. Build	2010. Revise Plan			█							
	2020. Synchronize Artifacts			█							
	2030. Analyze			█							
	2031. Define Essential Use Cases			█	█						
	2032. Refine Use Case Diagrams			█	█						
	2033. Define Domain Model			█	█						
	2034. Refine Glossary			█	█						
	2035. Define System Sequence Diagrams			█	█	█					
	2036. Define Operation Contracts			█	█	█					
	2037. Define State Diagrams			█	█	█					
	2038. Refine System Test Case			█	█	█					
	2039. Analyze (2030) Traceability Analysis			█	█	█					
	2040. Design			█	█	█	█				
	2041. Design Real Use Cases			█	█	█	█				
	2042. Define Reports, UI, and Storyboards			█	█	█	█				
	2043. Refine System Architecture			█	█	█	█				
	2044. Define Interaction Diagrams			█	█	█	█				
	2045. Define Design Class Diagrams			█	█	█	█				
	2046. Design Traceability Analysis			█	█	█	█				
	2047. Define Database Schema			█	█	█	█				
	2050. Construct			█	█	█	█	█			
	2051. Implement Class & Methods Definitions			█	█	█	█	█	█		
	2052. Implements Windows			█	█	█	█	█	█		
	2053. Implement Reports			█	█	█	█	█	█		
	2054. Implement DB Schema			█	█	█	█	█	█		
	2055. Write Unit Test Code			█	█	█	█	█	█		
	2060. Test			█	█	█	█	█	█		
2061. Unit Testing			█	█	█	█	█	█			
2062. Integration Testing			█	█	█	█	█	█			
2063. System Testing			█	█	█	█	█	█			
2064. Performance Testing			█	█	█	█	█	█			
2065. Acceptance Testing			█	█	█	█	█	█			
2066. Documentation Testing			█	█	█	█	█	█			
2067. Testing Traceability Analysis			█	█	█	█	█	█			