Software Modeling & Analysis

One More Chance

OOPT stage 1000 ver.3

Plan & Elaboration

Team No	Team2
과목	Software Modeling and Analysis
담당교수	JUNBEOM YOO Associate Professor / Ph.D
팀 구성원	201014184 김도윤
	201111367 여승훈
	201111347 김태호
제출일자	2016-03-31

Index

Activity1001. **Define Draft Plan**

Activity1002. **Create Preliminary Investigation Report**

Activity1003. **Define Requirements**

Activity1004. **Record Terms in Glossary**

Activity1005. **Implement Prototype**

Activity1006. **Define Draft System Architecture**

Activity1007. **Define Business Use Case**

Activity1008. **Define Business Concept Model**

Activity1009. **Define System Test Case**

Activity1010. **Refine Plan**

Activity 1001. Define Draft Plan

1. Motivation

소프트웨어의 개발이 활발해짐에 따라 소스코드의 복제, 도용이 증가하고 있다. 따라서 소스코드 복제 여부에 대한 검사 또한 매우 중요해지고 있다. 소스코드가 복제되면 기업, 개인 간의 분쟁이 일어날 수 있다. 따라서 사전에 소프트웨어 유사도 검사가 필요하다. 이것은 학교에서 과제로 진행되는 프로젝트에도 해당되며, 배움에 있어 학생들의 소스코드 복제 및 도용은 옳지 않다. 따라서 Clone Checker 프로그램을 개발하게 되었다.

2. Project Objectives

다수의 C 프로그램을 대장으로 상호 cheating 여부를 정량적으로 판단하고, 해당 내용을 다시 한 번 검토하여 결과를 수정할 수 있는 Clone Checker

- 코드 유사성의 단순 수치화를 통해 코드의 유사성을 직관적으로 판단할 수 있다.
- 각 코드들간의 유사성을 등급으로 표시하여 알려준다.
- 복제 및 도용이 의심되는 코드를 User가 다시 한번 확인하여 판단할 수 있다.

3. Project Scope

'One more chance'는 program이 산출한 결과에 의해 검사대상이 억울한 피해를 받을 수 있는 가능성이 존재한다는 사실에 착안하였다. 이러한 부분에 대해 다시 한 번 확인할 기회를 주는 것을 목표로 한다.

4. Functional requirements

- Load Code
- Delete Code
- Display Code List
- Detect Duplicate Code
- Clone Check
- Check Value
- Check Operation
- Calculate Similarity
- Display Code Set List
- Show All
- Show Red
- Show Orange
- Show Yellow
- Show Green
- Select Code Set
- Display Main Code
- Display Sub Code
- Display Similarity
- User Accept
- User Reject
- Save

5. Non-functional requirements

- 검사시간을 3min/50file 이내로 유지한다.

6. Resource Estimation

6.1 Human efforts (M/M)

3M/3M

6.2 Human resources

3명 – 프로그래머 3명

6.3 Duration

3개월

6.4 Budget

T.F Wage : 30 * 3man * 3month = 270

7. Other Information

- Future Version

  향상된 Performance, 한번이라도 Clone Checker 프로그램을 통해 객체화된 Code는 정보가 살아있어 다음 Clone Check에 이용된다.

Activity 1002. Create Preliminary Investigation Report

1. Alternative solution

1.1 시중에 판매하는 Clone Check Program을 구매한다.

1.2 인력을 동원한다.

1.3 개발업체에 제작을 의뢰한다.

2. Project justification

1) Cost : 2,700,000₩

2) Duration : 3 Month

3) Risk : OOPT에 대한 이해력 부족. Clone Checker에 대한 지식 부족, 과제 Stack Overflow, 벚꽃사랑. 팀원의 휴학

4) Effect : 프로그램 유지 보수가 수월하다. 양산되는 과제물에 대한 Check가 용이하다.

비용 측면에서 저렴하다.

3. Risk & Risk reduction plan

3.1 Risk Management

Risk	Probability	Significance	Weight
Un-Skilled OOPT	3	5	15
Lack of knowledge Clone Checker	2	4	8
Assignment Stack Overflow	5	2	10
Spring Love	1	5	5
Absence	1	10	10

3.2 Risk Reduction Plan

Risk	Reduction Plan
Un-Skilled OOPT	교수님 혹은 강의조교에게 자문을 구한다.
Lack of knowledge Clone Checker	Clone Checker 구동 방식에 대해 공부한다.
Assignment Stack Overflow	Time Sharing 기법을 활용하여 동시에 처리한다.
Spring Love	합숙 프로젝트를 진행한다. (가능성 배제)
Absence	설득한다

4. Market Analysis

- Alphago에 사회적 issue화에 의해 증가할 것으로 전망되는 Computer Engineering 시장.

5. Managerial issue

- 2016년 5월까지 proto-type 개발이 완료되어야 한다.
- 2016년 7월까지 Final version 개발이 완료되어야 한다.

Activity 1003. Define Requirements

1. Functional Requirements

Function	Description
Load Code	검사할 Code를 List에 Load한다.
Delete Code	List에 올라와있던 Code를 List에서 삭제한다.
Display Code List	Load된 Code의 List를 보여준다.
Detect Duplicate Code	Load된 모든 Code간의 Duplicate 여부를 Detect 시작한다..
Clone Check	Load된 모든 Code간의 Duplicate 여부를 Detect 한다.
Check Value	모든 Code간의 Value 유사도 검사를 실행한다.
Check Operation	모든 Code간의 Operation 유사도 검사를 실행한다.
Calculate Similarity	모든 Code간의 유사성을 계산한다.
Display Code Set List	Code Set List를 display한다.
Show All	Clone Check 후의 모든 Code Set의 유사성을 표시한다.
Show Red	Clone Check 후 두 코드간 유사성이 100~90%인 Code Set을 표시한다.
Show Orange	Clone Check 후 두 코드간 유사성이 90~70%인 Code Set을 표시한다
Show Yellow	Clone Check 후 두 코드간 유사성이 70~50%인 Code Set을 표시한다
Show Green	Clone Check 후 두 코드간 유사성이 50% 미만인 Code Set을 표시한다
Select Code Set	Check List에서 Code Set을 선택한다.
Display Main Code	List에서 선택한 비교항목의 Main Code를 display한다.
Display Sub Code	List에서 선택한 비교항목의 Sub Code를 display한다.
Display Similarity	선택한 두 Code의 Similarity를 Progress Bar에 Display한다.
User Accept	User가 선택한 두 Code의 Duplicate가 의심되는 구간을 확인하고 Clone으로 판단한다.
User Reject	User가 선택한 두 Code의 Duplicate가 의심되는 구간을 확인하고 Clone이 아니라고 판단한다.
Save	User가 검토한 정보를 저장하여 list에 적용한다.

2. Functional Requirements (Categorized Table)

Reference No.	Function	Category
R1.1	Load Code	Evident
R1.2	Delete Code	Evident
R1.3	Display Code List	Hidden
R2.1.1	Detect Duplicate Code	Evident
R2.1.2	Clone Check	Hidden
R2.1.3	Check Value	Hidden
R2.1.4	Check Operation	Hidden
R2.1.5	Calculate Similarity	Hidden
R2.1.6	Display Code Set List	Hidden
R3.1	Show All	Evident
R3.2	Show Red	Evident
R3.3	Show Orange	Evident
R3.4	Show Yellow	Evident
R3.5	Show Green	Evident
R4.1	Select Code Set	Evident
R4.2	Display Main Code	Hidden
R4.3	Display Sub Code	Hidden
R4.4	Display Similarity	Hidden
R5.1	User Accept	Evident
R5.2	User Reject	Evident
R6	Save	Evident

3. Performance Requirements

- 검사시간은 3min/50File 이내로 유지되어야 한다.

4. Operating Environment

Source	Case1	Case2	Case3
Operating Software	Win7 – 64bit	Win8 – 64bit	Mac OS – 64bit
CPU	i3-4005U	i5-4200U	i5-4200U
RAM	8GB	4GB	8GB
Library	128SSD	128GB SSD	128GB SSD

Develop Environment: Eclipse LUNA/MARS/KEPLER

5. Interface Requirements

- 유사도 percent에 따른 등급을 보기 좋게 표현해야 한다.
- 유사도 등급 기준에 따라 Code Set이 정확하게 분리되어야 한다.
- Duplicate가 의심되는 부분을 출력하여 User가 판단할 수 있도록 한다.
- User가 유사도 그래프를 확인하여 직관적인 관찰이 가능하도록 한다.

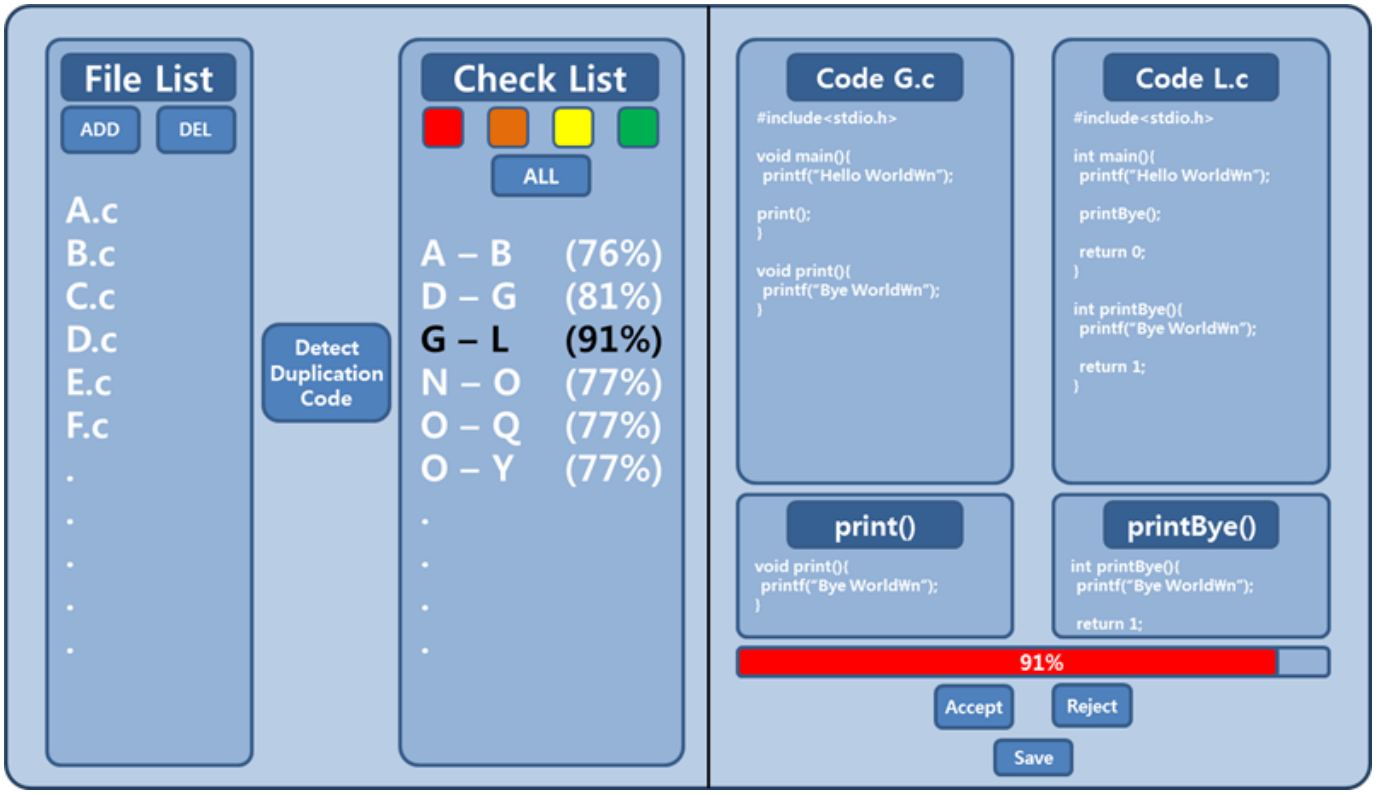
6. Other Requirements

- N/A

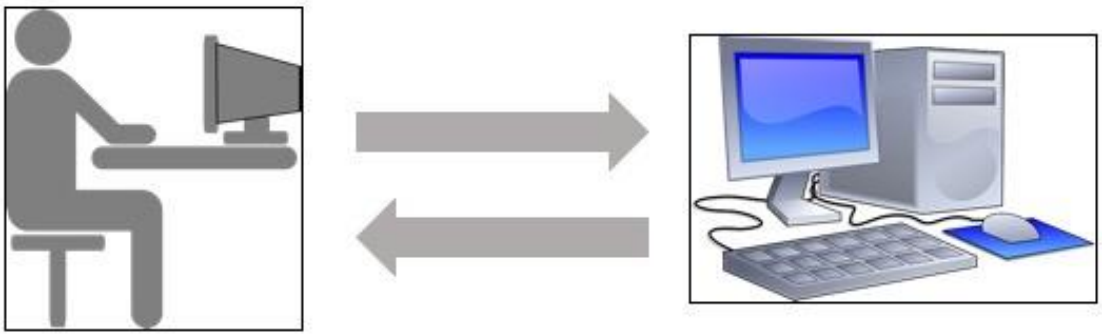
Activity 1004. Record Terms in Glossary

Term	Description	Remarks
User	Clone Checker 프로그램을 사용하여 Clone 여부를 판단하는 객체	
Value	Code 에서 선언된 Value 의 객체	
Operation	Code 에서 선언된 Operation 의 객체	
Branch	Code 에서 선언된 Branch 의 객체	
Code	User 가 Load 한 Code 객체	
Code Set	검사가 끝난 Code Set 객체	
Main Controller	프로그램을 제어하는 객체	

Activity 1005. Implement Prototype

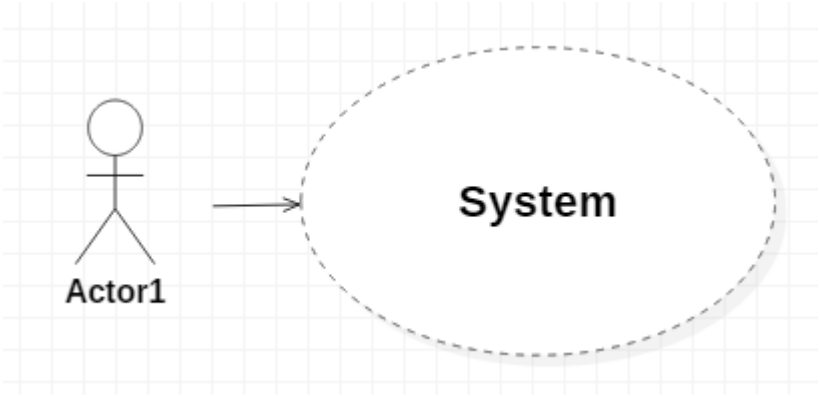


Activity 1006. Define Draft System Architecture



Activity 1007. Define Business Use Case

1. Define system boundary

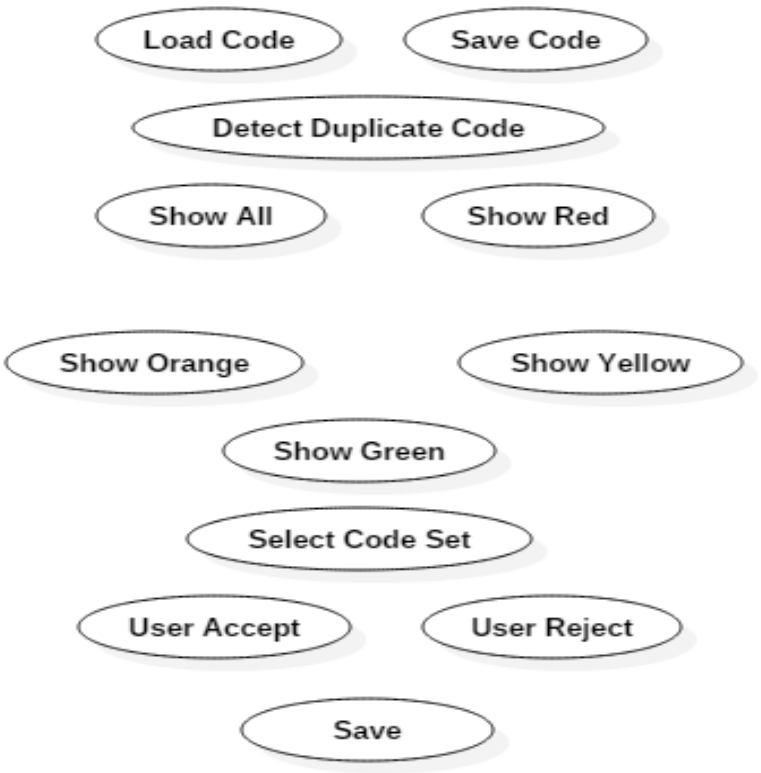


2. Identify and describe actors

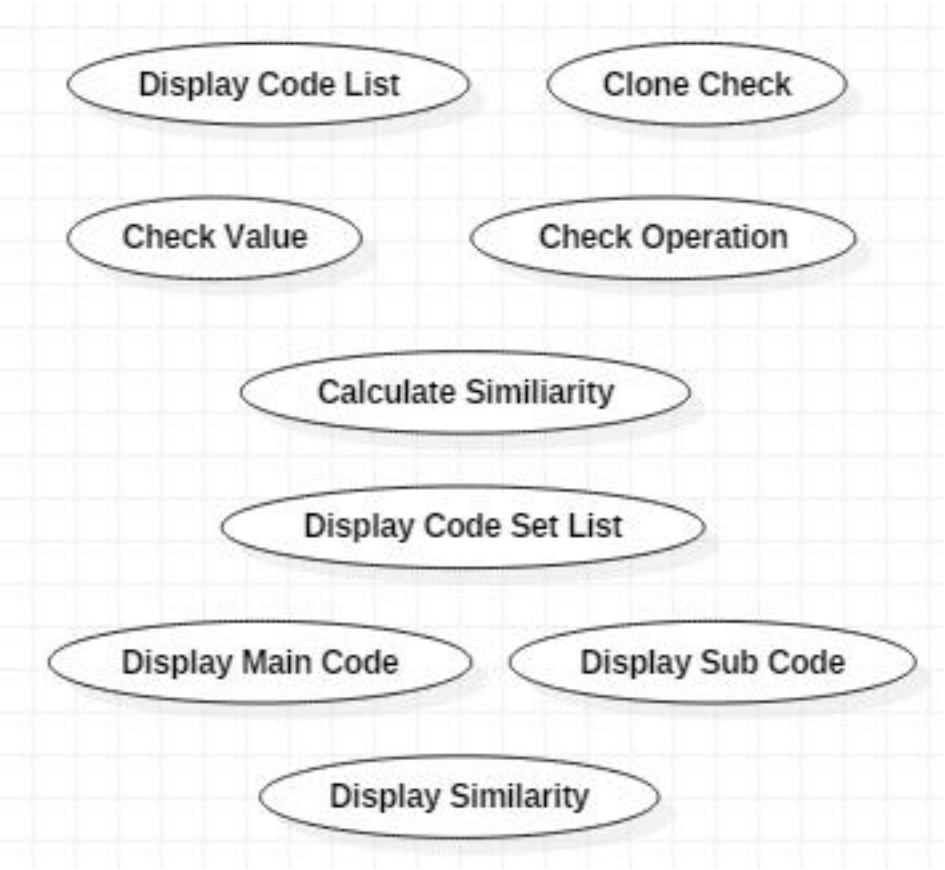
- User : Duplicate Code를 Detect하는 대상.

3. Identify use cases

3.1 Actor-Based



3.2 Event-Based



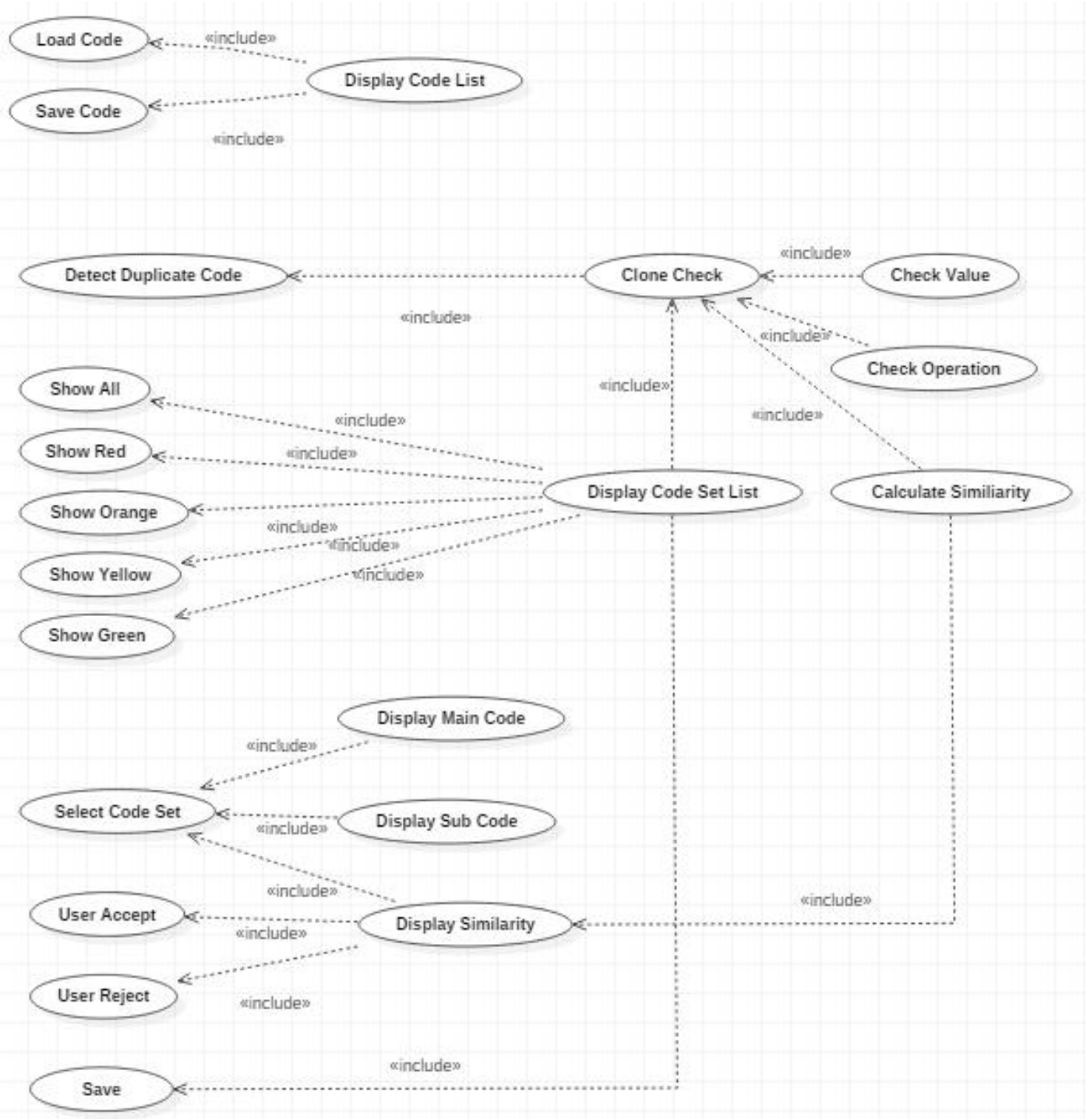
4. Allocate system functions into related use cases

Ref. #.	Function	Use Case Number & Name	Category
R1.1	Load Code	1. Load Code	
R1.2	Delete Code	2. Delete Code	
R1.3	Display Code List	3. Display Code List	
R2.1.1	Detect Duplicate Code	4. Detect Duplicate Code	
R2.1.2	Clone Check	5. Clone Check	
R2.1.3	Check Value	6. Check Value	
R2.1.4	Check Operation	7. Check Operation	
R2.1.5	Calculate Similarity	8. Calculate Similarity	
R2.1.6	Display Code Set List	9. Display Code Set List	
R3.1	Show All	10. Show All	
R3.2	Show Red	11. Show Red	
R3.3	Show Orange	12. Show Orange	
R3.4	Show Yellow	13. Show Yellow	
R3.5	Show Green	14. Show Green	
R4.1	Select Code Set	15. Select Code Set	
R4.2	Display Main Code	16. Display Main Code	
R4.3	Display Sub Code	17. Display Sub Code	
R4.4	Display Similarity	18. Display Similarity	
R5.1	User Accept	19. User Accept	
R5.2	User Reject	20. User Reject	
R6	Save	21. Save	

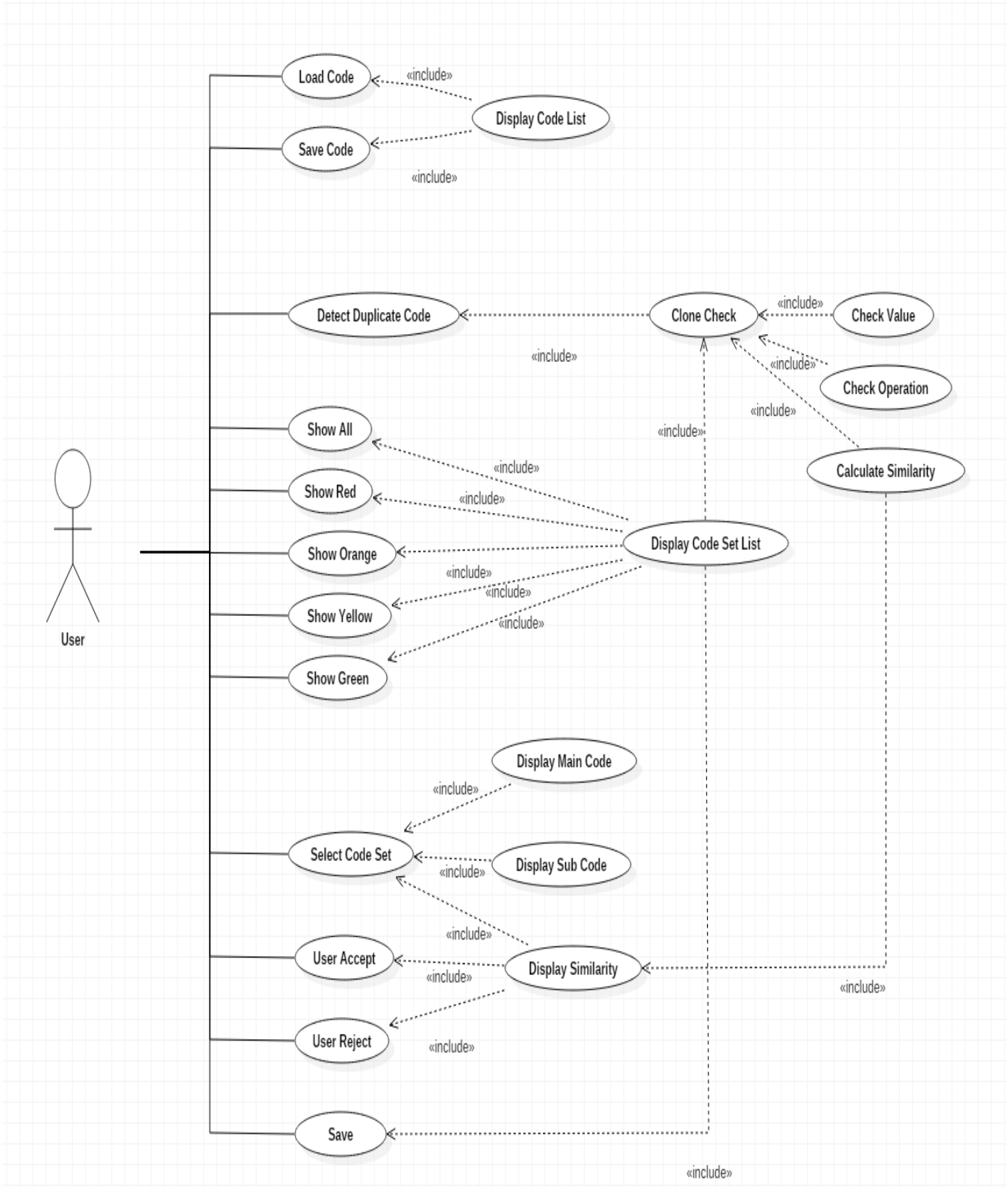
5. Categorize use cases

Ref. #.	Function	Use Case Number & Name	Category
R1.1	Load Code	1. Load Code	Primary
R1.2	Delete Code	2. Delete Code	Primary
R1.3	Display Code List	3. Display Code List	Primary
R2.1.1	Detect Duplicate Code	4. Detect Duplicate Code	Primary
R2.1.2	Clone Check	5. Clone Check	Primary
R2.1.3	Check Value	6. Check Value	Primary
R2.1.4	Check Operation	7. Check Operation	Primary
R2.1.5	Calculate Similarity	8. Calculate Similarity	Primary
R2.1.6	Display Code Set List	9. Display Code Set List	Primary
R3.1	Show All	10. Show All	Primary
R3.2	Show Red	11. Show Red	Primary
R3.3	Show Orange	12. Show Orange	Primary
R3.4	Show Yellow	13. Show Yellow	Primary
R3.5	Show Green	14. Show Green	Primary
R4.1	Select Code Set	15. Select Code Set	Primary
R4.2	Display Main Code	16. Display Main Code	Primary
R4.3	Display Sub Code	17. Display Sub Code	Primary
R4.4	Display Similarity	18. Display Similarity	Primary
R5.1	User Accept	19. User Accept	Primary
R5.2	User Reject	20. User Reject	Primary
R6	Save	21. Save	Primary

6. Identify relationships between use cases



7. Draw a use case diagram



8. Describe use cases

Use Case	1. Load Code
Actors	User
Description	검사할 Code를 List에 Load한다.

Use Case	2. Delete Code
Actors	User
Description	List에 올라와있던 Code를 List에서 삭제한다.

Use Case	3. Display Code List
Actors	None
Description	Load된 Code의 List를 보여준다.

Use Case	4. Detect Duplicate Code
Actors	User
Description	Load된 모든 Code간의 Duplicate 여부를 Detect 시작한다.

Use Case	5. Clone Check
Actors	User
Description	Load된 모든 Code간의 Duplicate 여부를 Detect한다.

Use Case	6. Check Value
Actors	None
Description	모든 Code간의 Value 유사도 검사를 실행한다.

Use Case	7. Check Operation
Actors	None
Description	모든 Code간의 Operation 유사도 검사를 실행한다.

Use Case	8. Calculate Similarity
Actors	None
Description	모든 Code간의 유사성을 계산한다.

Use Case	9. Display Code Set List
Actors	None
Description	Code Set List를 Display한다.

Use Case	10. Show All
Actors	User
Description	Clone Check 후의 모든 Code Set의 유사성을 표시한다.

Use Case	11. Show Red
Actors	User
Description	Clone Check 후 두 코드간 유사성이 100~90%인 Code Set을 표시한다.

Use Case	12. Show Orange
Actors	User
Description	Clone Check 후 두 코드간 유사성이 90~70%인 Code Set을 표시한다.

Use Case	13. Show Yellow
Actors	User
Description	Clone Check 후 두 코드간 유사성이 70~50%인 Code Set을 표시한다.

Use Case	14. Show Green
Actors	User
Description	Clone Check 후 두 코드간 유사성이 50% 미만인 Code Set을 표시한다.

Use Case	15. Select Code Set
Actors	User
Description	Check List에서 Code Set을 선택한다.

Use Case	16. Display Main Code
Actors	None
Description	List에서 선택한 비교항목의 Main Code를 display한다.

Use Case	17. Display Sub Code
Actors	None
Description	List에서 선택한 비교항목의 Sub Code를 display한다.

Use Case	18. Display Similarity
Actors	None
Description	선택한 두 Code의 Similarity를 Progress Bar에 Display한다.

Use Case	19. User Accept
Actors	User
Description	User가 선택한 두 Code의 Duplicate가 의심되는 구간을 확인하고 Clone으로 판단한다.

Use Case	20. User Reject
Actors	User
Description	User가 선택한 두 Code의 Duplicate가 의심되는 구간을 확인하고 Clone이 아니라고 판단한다.

Use Case	21. Save
Actors	User
Description	User가 검토한 정보를 저장하여 list에 적용한다.

9. Rank use cases

Ref. #.	Function	Use Case Number & Name	Category	Rank
R1.1	Load Code	1. Load Code	Primary	High
R1.2	Delete Code	2. Delete Code	Primary	High
R1.3	Display Code List	3. Display Code List	Primary	High
R2.1.1	Detect Duplicate Code	4. Detect Duplicate Code	Primary	High
R2.1.2	Clone Check	5. Clone Check	Primary	High
R2.1.3	Check Value	6. Check Value	Primary	High
R2.1.4	Check Operation	7. Check Operation	Primary	High
R2.1.5	Calculate Similarity	8. Calculate Similarity	Primary	High
R2.1.6	Display Code Set List	9. Display Code Set List	Primary	High
R3.1	Show All	10. Show All	Primary	High
R3.2	Show Red	11. Show Red	Primary	High
R3.3	Show Orange	12. Show Orange	Primary	High
R3.4	Show Yellow	13. Show Yellow	Primary	High
R3.5	Show Green	14. Show Green	Primary	High
R4.1	Select Code Set	15. Select Code Set	Primary	High
R4.2	Display Main Code	16. Display Main Code	Primary	High
R4.3	Display Sub Code	17. Display Sub Code	Primary	High
R4.4	Display Similarity	18. Display Similarity	Primary	High
R5.1	User Accept	19. User Accept	Primary	High
R5.2	User Reject	20. User Reject	Primary	High
R6	Save	21. Save	Primary	High

Activity 1008. Define Business Concept Model

Activity 1009. Define System Test Case

Test Number	Test 항목	Description	Use-Case	System Function
1	Load 버튼 시험	User가 Load한 Code File이 List에 Load되는지 확인한다.	1. Load Code	R1.1
2	Delete 버튼시험	User가 Delete한 Code File이 List에서 Delete되는지 확인한다.	2. Delete Code	R1.2
3	Display 시험	User가 Load한 List가 Display 되는지 확인한다.	3. Display Code List	R1.3
4	Detect 버튼시험	User가 요청한 Detect Duplicate Code 명령이 수행되는지 확인한다.	4. Detect Duplicate Code	R2.1.1
5	Clone Check 여부 시험	Load된 모든 Code간의 Clone Check가 수행되는지 확인한다.	5. Clone Check	R2.1.2
6	Check Value 시험	Code들간의 Value 유사도 검사가 수행되는지 확인한다.	6. Check Value	R2.1.3
7	Check Operation 시험	Code들간의 Operation 유사도 검사가 수행되는지 확인한다.	7. Check Operation	R2.1.4
8	Calculate Similarity 시험	Code간의 유사성이 계산되는지 확인한다.	8. Calculate Similarity	R2.1.5
9	Code Set Display 시험	Code Set List들이 Display 되는지 확인한다.	9. Display Code Set List	R2.1.6
10	All 버튼 시험	모든 Code Set List가 Display 되는지 확인한다.	10. Show All	R3.1
11	Red 버튼 시험	코드간 유사성이 100~90%인 Code Set이 Display 되는지 확인한다.	11. Show Red	R3.2
12	Orange 버튼 시험	코드간 유사성이 90~70%인 Code Set이 Display 되는지 확인한다	12. Show Orange	R3.3
13	Yellow 버튼 시험	코드간 유사성이 70~50%인 Code Set이 Display 되는지 확인한다	13. Show Yellow	R3.4
14	Green 버튼 시험	코드간 유사성이 50% 미만인 Code Set이 Display 되는지 확인한다	14. Show Green	R3.5

15	Code Set Select 시험	Check List에서 Code Set이. 선택되는지 확인한다.	15. Select Code Set	R4.1
16	Main Code Display 시험	List에서 선택한 비교항목의 Main Code가 Display 되는지 확인한다.	16. Display Main Code	R4.2
17	Sub Code Display 시험	List에서 선택한 비교항목의 Sub Code가 Display 되는지 확인한다.	17. Display Sub Code	R4.3
18	Progress Bar Display 시험	선택한 두 Code의 Similarity가 Progress Bar에 Display 되는지 확인한다.	18. Display Similarity	R4.4
19	Accept 버튼 시험	User Accept 명령이 정확하게 전달되는지 확인한다.	19. User Accept	R5.1
20	Reject 버튼 시험	User의 Reject 명령이 정확하게 전달되는지 확인한다.	20. User Reject	R5.2
21	Save 버튼 시험	User의 Save 명령이 정확하게 전달되는지 확인한다.	21. Save	R6

Activity 1010. Refine Plan

1. Project Scope

소스코드의 복제 및 도용으로 인한 문제를 방지하기 위해 소스코드간의 Cloning 여부를 판단한다.

2. Project Objectives

다수의 C 프로그램을 대장으로 상호 cheating 여부를 정량적으로 판단하고, 해당 내용을 User가 검토하여 결과를 수정할 수 있는 Clone Checker를 개발한다.

3. Functional Requirements

- Load Code
- Delete Code
- Display Code List
- Detect Duplicate Code
- Clone Check
- Check Value
- Check Operation
- Calculate Similarity
- Display Code Set List
- Show All
- Show Red
- Show Orange
- Show Yellow
- Show Green
- Select Code Set
- Display Main Code
- Display Sub Code
- Display Similarity

- User Accept
- User Reject
- Save

4. Performance Requirements

- 검사시간을 3min/50File 이내로 유지한다.

5. Operating Environment

Source	Case1	Case2	Case3
Operating Software	Win7 – 64bit	Win8 – 64bit	Mac OS – 64bit
CPU	i3-4005U	i5-4200U	i5-4200U
RAM	8GB	4GB	8GB
Library	128SSD	128GB SSD	128GB SSD

6. User Interface Requirements

- 유사도 percent에 따른 등급을 보기 좋게 표현해야 한다.
- 유사도 등급 기준에 따라 Code Set이 정확하게 분리되어야 한다.
- Duplicate가 의심되는 부분을 출력하여 User가 판단할 수 있도록 한다.
- User가 유사도 그래프를 확인하여 직관적인 관찰이 가능하도록 한다.

7. Other Requirements

- N/A

8. Resources

8.1 Human efforts (M/M)

3M/3M

8.2 Human resources

3명 – 프로그래머 3명

8.3 Duration

3개월

8.4 Budget

₩2,700,000

9. Scheduling

Stage	Phase(00X0)/Activity(000X)	Schedule(Week) 1	2	3	4	5	6	7	8	9	10
1000. Plan & Elaborate	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	2110. Revise Plan			■	■						
	2120. Synchronize Artifacts			■	■						
	2130. Analyze										
	2131. Define Essential Use Case					■					
	2132. Refine Use Case Diagram					■					
	2133. Refine Conceptual Model					■					
	2134. Refine Glossary					■					
	2135. Define System Sequence Diagram					■	■				
	2136. Define Operation Contracts					■	■				
	2137. Define State Diagrams					■	■				
	2140. Design										
	2141. Define Real Use Case						■				
	2142. Define Reports, UI and Story boards						■				
	2143. Refine System Architecture						■				
	2144. Define Interaction Diagram						■				
	2145. Define Design Class Diagrams						■	■			
	2150. Construct										
	2151. Implement Class & Interface Definition							■			
	2152. Implement Methods							■			
	2153. Implement Windows							■	■		
	2154. Implement Reports							■	■		
	2155. Write Test Code							■	■		
	2160. Test										
	2161. Unit Testing									■	
	2162. Integration Testing									■	■
	2163. System Testing										■
	2164. Performance Testing										■
	2165. Acceptance Testing										■
	2166. Documentation Testing										■