Clone Check

with User

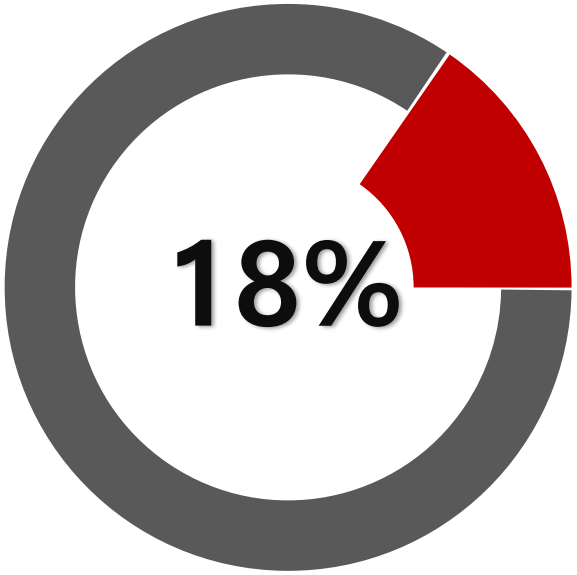
[SMA][T2]
2010 14184 김도윤
2011 11367 여승훈
2011 11347 김태호

Index

1 현황 분석

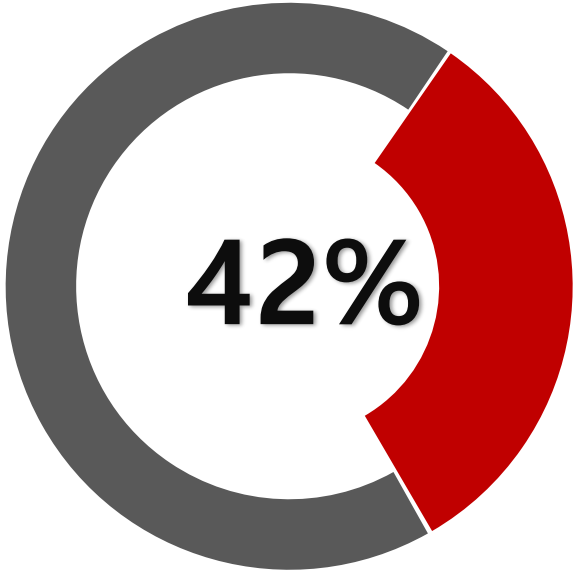
User Wants

침해 비율



총 1,112건 중 **242건**

침해금액 비율



총 ₩303억 중 **₩125억**

출처 : 인터넷 법률신문뉴스 '검찰, 불법복제 SW사용 업체 무더기 적발'

1 현황 분석

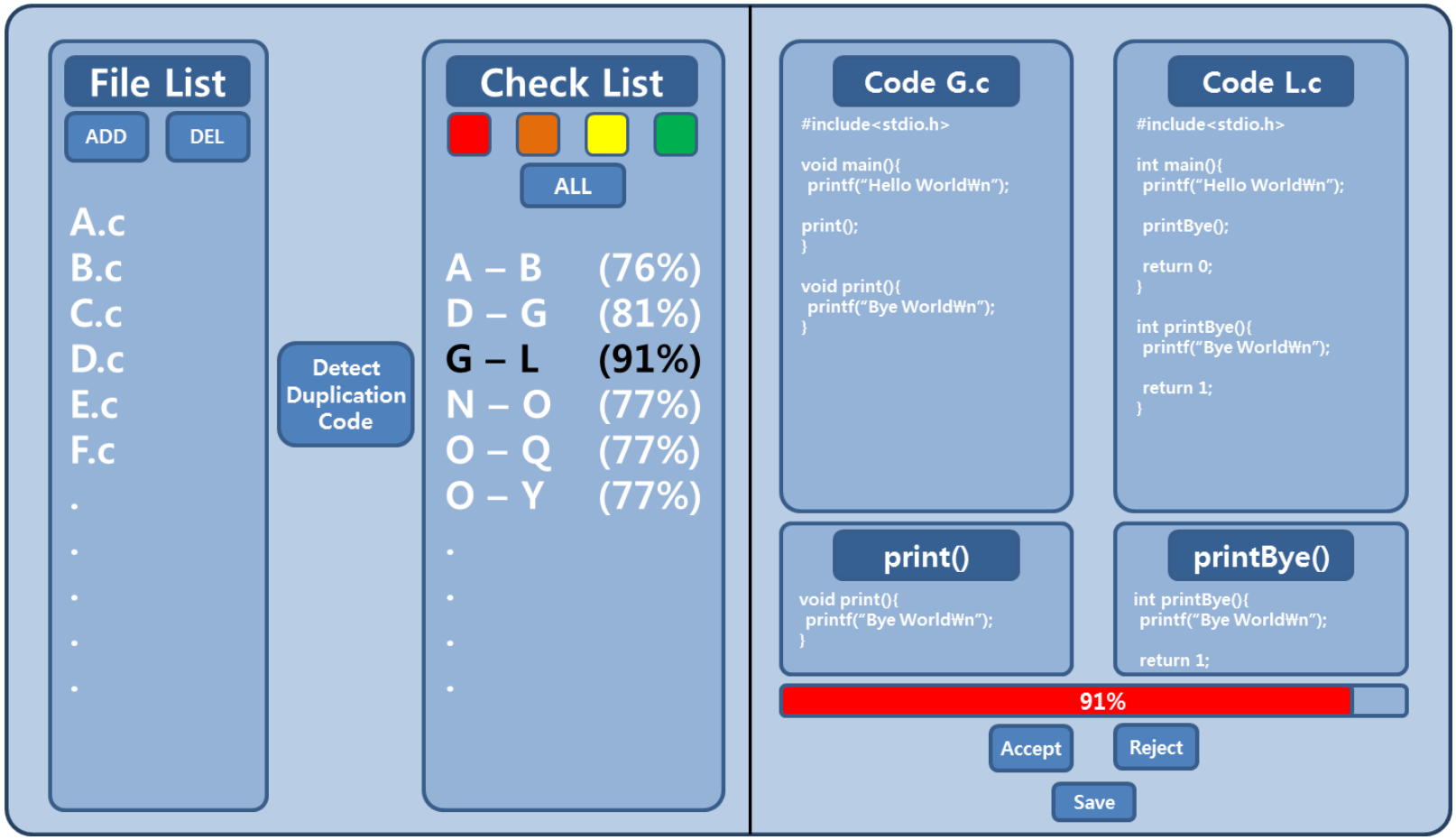
Market Analyze

1 현황 분석

Market Analyze

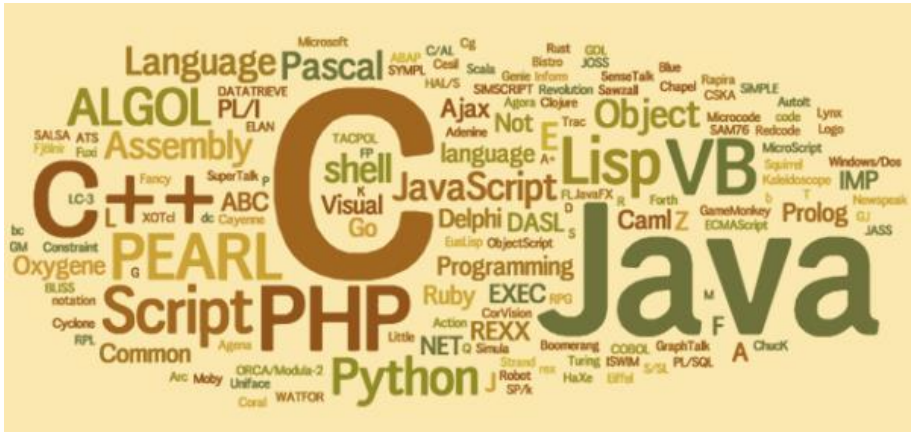
2 제품 설명

User Interface & Process



2 제품 설명

Expected Effect



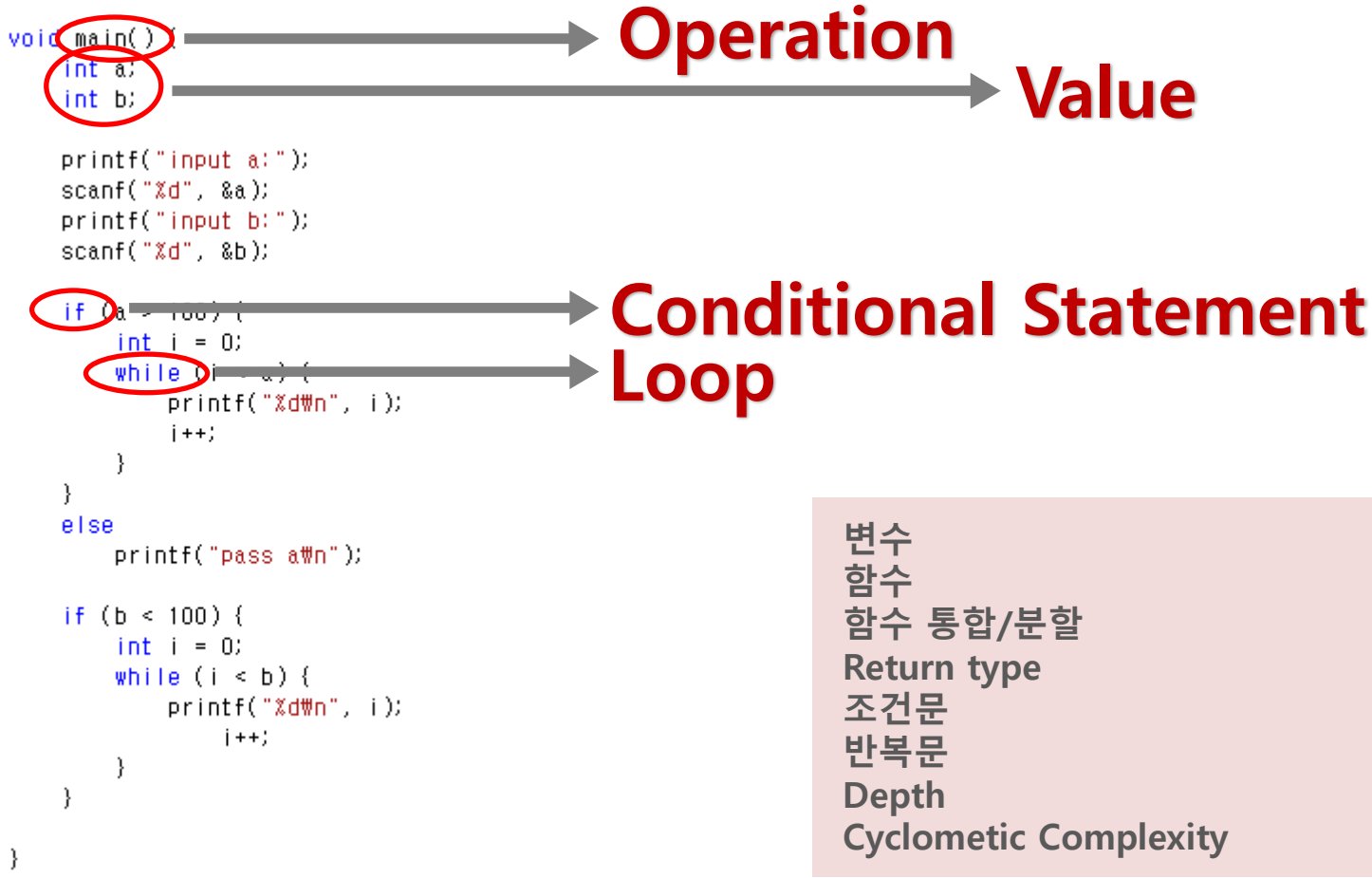
3 시스템 설명

Functional Requirement

Function	Description
Load Code	검사할 Code를 Load한다.
Delete Code	List에 올라와있던 Code를 Code 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.
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.

3 시스템 설명

Solution

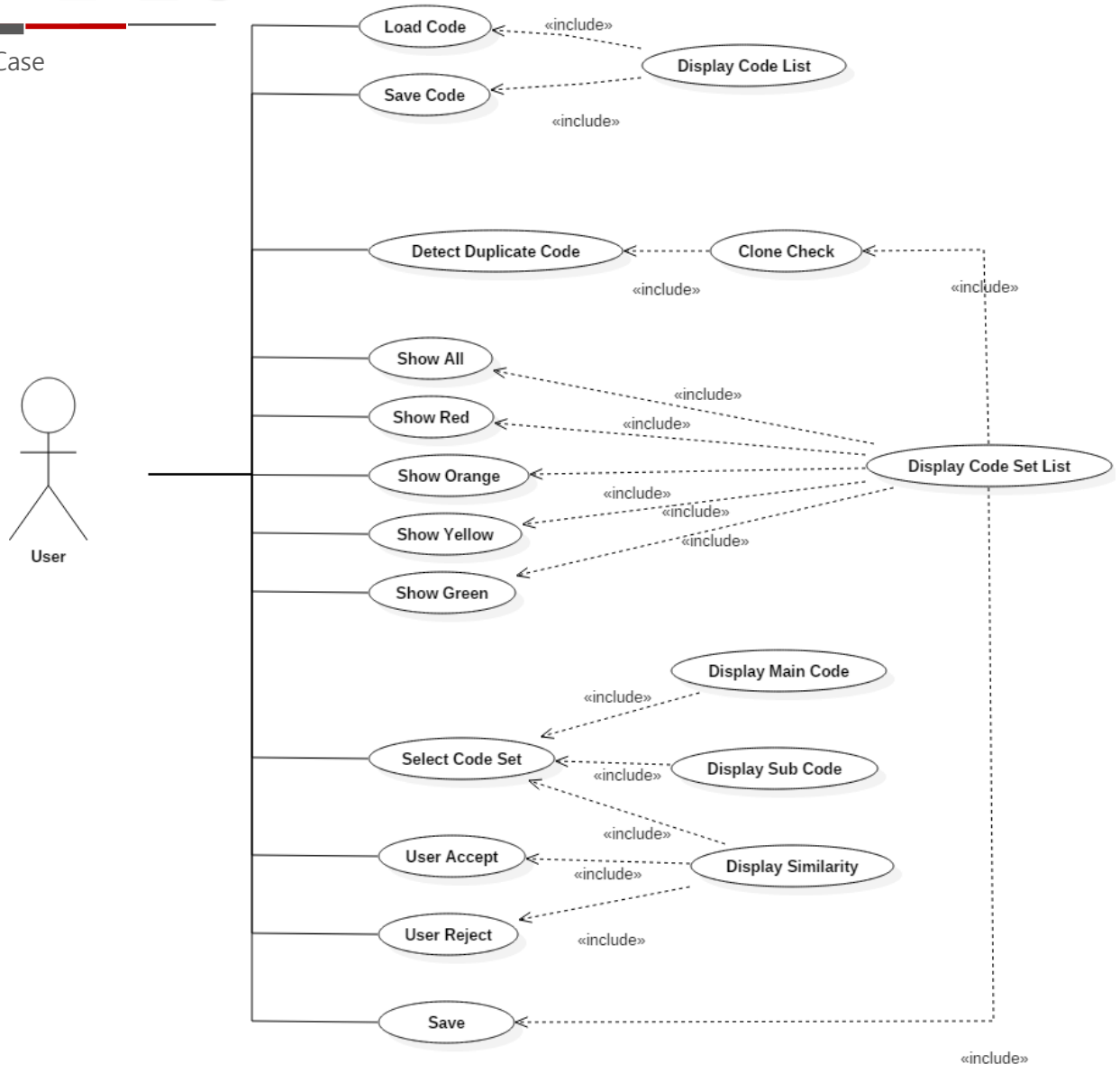



```
void main() {
    int a;
    int b;

    printf("input a:");
    scanf("%d", &a);
    printf("input b:");
    scanf("%d", &b);

    if (a < 100) {
        int i = 0;
        while (i < a) {
            printf("%d₩n", i);
            i++;
        }
    }
    else
        printf("pass a₩n");

    if (b < 100) {
        int i = 0;
        while (i < b) {
            printf("%d₩n", i);
                i++;
        }
    }

}
```

변수
함수
함수 통합/분할
Return type
조건문
반복문
Depth
Cyclometic Complexity

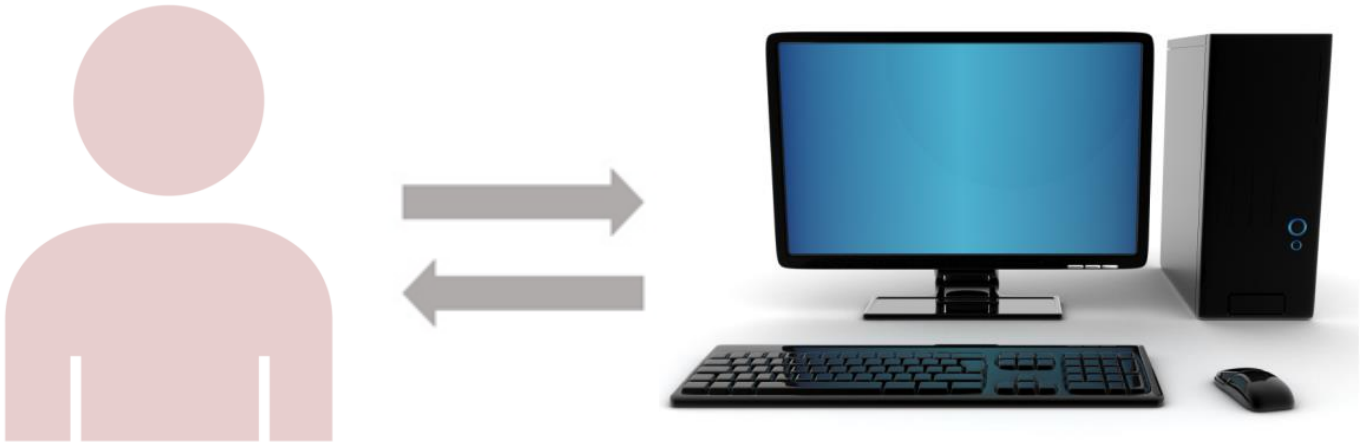
3 시스템 설명

Use-Case



3 시스템 설명

System Architecture

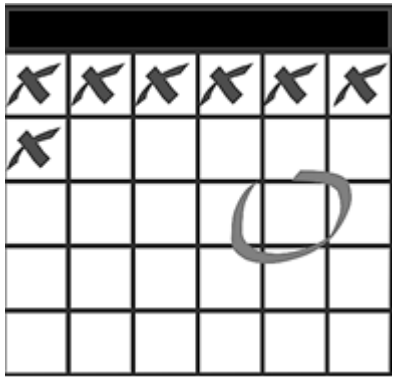


4 진행 계획

Project Resource

300,000₩/Month

3 Man



3 Month

4 진행 계획

Risk & Risk Reduction Plan

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**

4 진행 계획

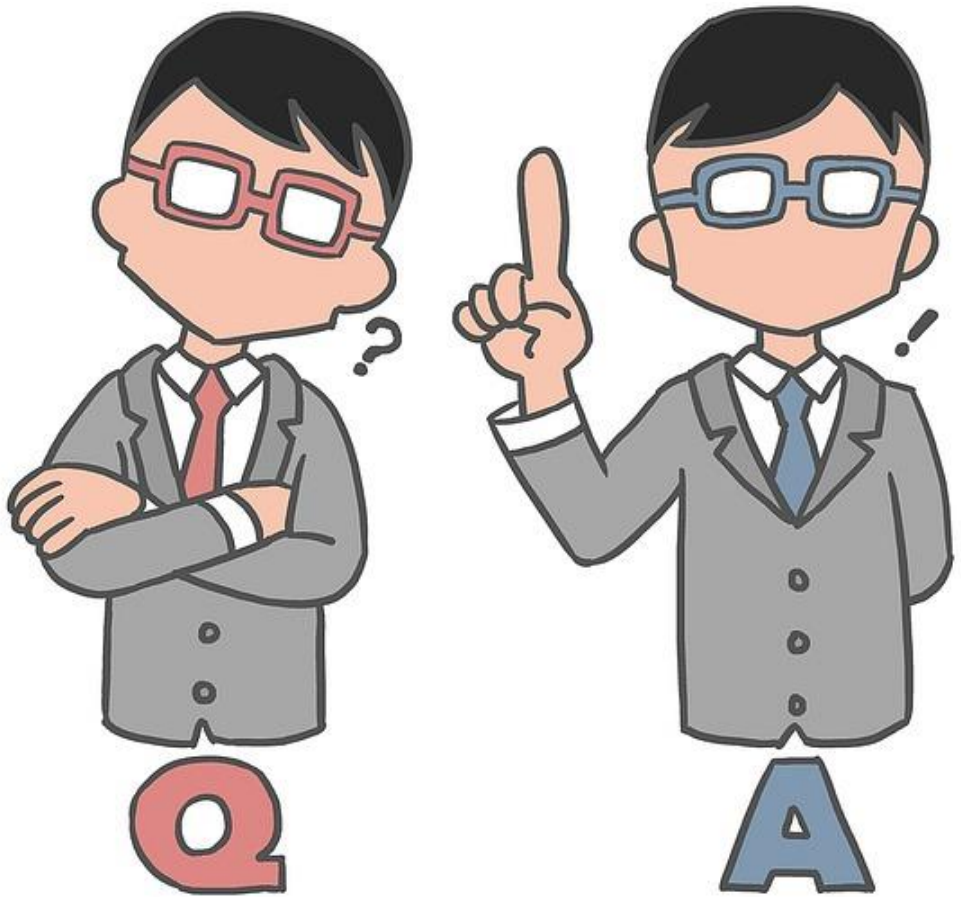
Risk & Risk Reduction Plan

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

4 진행 계획

Project Schedule

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										■


Q
A