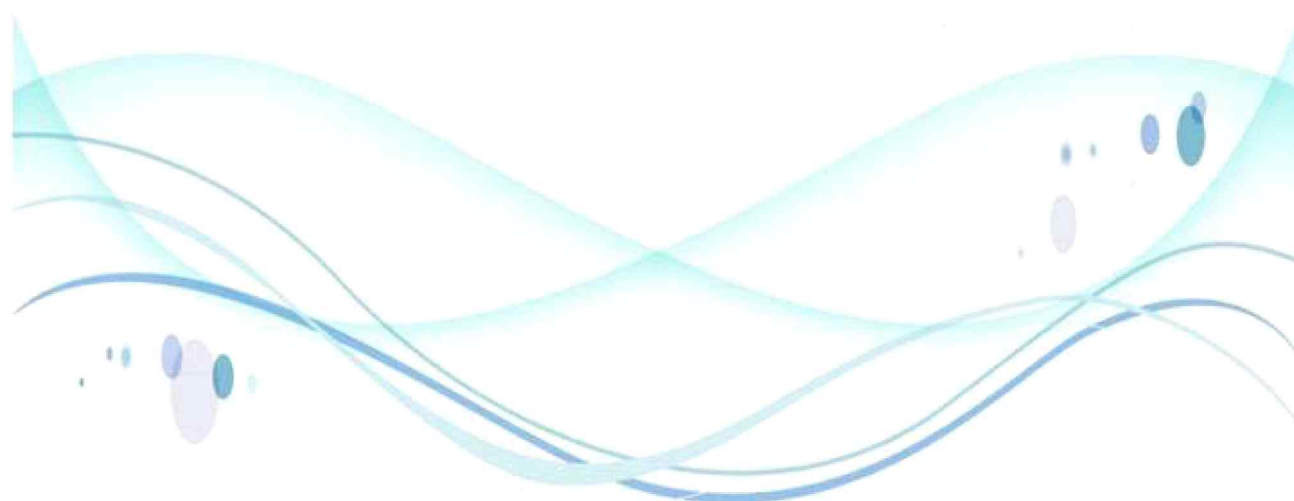




# Global Creative Software

**2015**

## **SW코드품질 측정과 분석**



# 목 차

I . 측정 및 분석 과정 소개 .....	2
II . 측정 및 분석 기술적 가이드 .....	3
1. 주요사항 .....	3
2. 측정 및 분석 절차 .....	6
3. 측정지표 .....	7

◆ 본 자료는 GCS(Global Creative Software) 수행기관을 대상으로 SW공학 관련 교육을 제공하는 GCS School의 교육 교재를 요약한 내용입니다.

- 과정명 : SW코드품질 측정과 분석
- 강사명 : 이종관 대표 (가온테스팅)

# I . 측정 및 분석 과정 소개

## 1 과정 개요

측정 및 분석이란 프로젝트 수행 중 필요한 정보를 수집하고 분석하여 프로젝트의 목표 달성 정도를 관리하기 위한 활동이다. 주요 활동을 측정하기 위한 목표와 항목을 선정하며, 계획에 따라 데이터를 수집한 후 내용을 분석하고, 그 결과를 관리한다.

### 1.1 학습 목표

측정 및 분석은 객관적인 의사결정과 향후 개선을 위해 필요한 정량적 활동의 수행 여부를 평가하기 위한 항목으로 정량적 데이터의 측정, 분석, 활용, 관리 활동을 포함한다. 측정지표에 대한 이해와 관리를 통해 현 상황에 대한 객관적 인식이 가능하며, 추후 개선에도 활용 할 수 있다. 본 교육은 특히 SW코드 품질을 측정하고 분석할 수 있도록 기술적 개념을 이해하는데 초점을 둔다.

### 1.2 기대 효과

- SW코드 품질에 대한 중요성을 인식한다.
- 품질 측정과 분석 프로세스를 이해한다.
- 품질 측정지표(Metrics) 전반에 대한 핵심개념을 이해하고 조직에 적용한다.
- 젠킨스(Jenkins) 및 소나큐브(SonarQube) 등의 품질 툴(도구)에서 제공되는 코드지표를 이해한다.
- 코드 커버리지(Coverage)의 종류 및 특성을 이해하고 조직의 특성에 맞추어 적용한다
- 코드 복잡도(Cyclomatic complexity metric)를 이해하고 적용한다.
- 코드 중복(Code Duplication)의 위험성을 인지한다.
- 정적 분석(Static Analysis)을 이해하고 정적분석을 통해 결함을 제거한다.

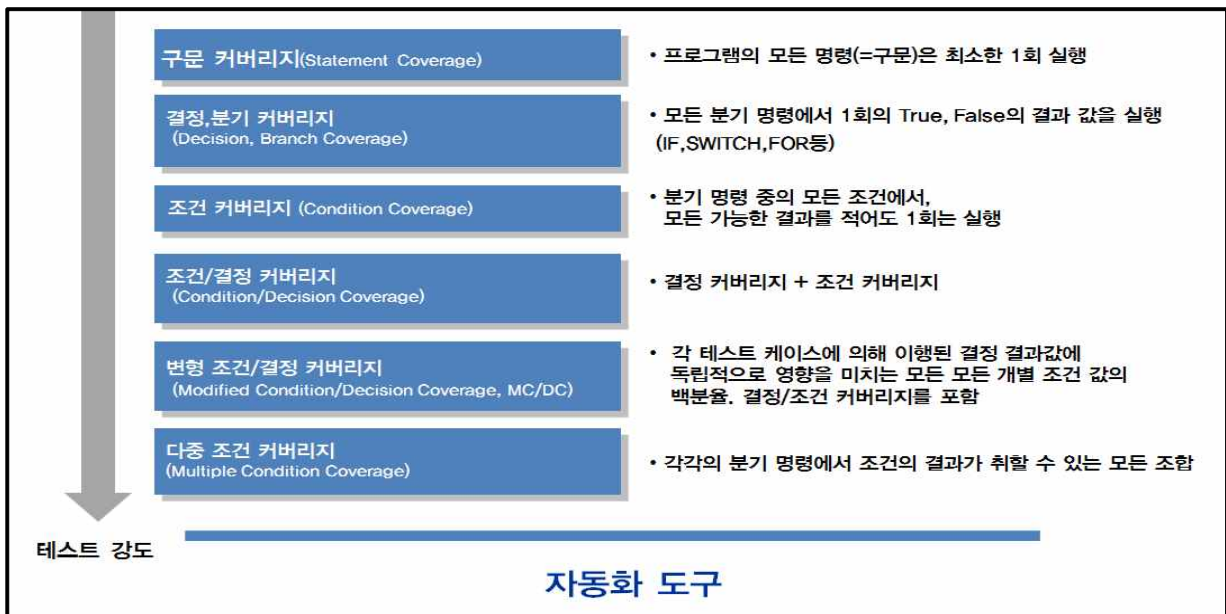
## II. 측정 및 분석 가이드

### 1 주요 사항

#### 1.1 코드 커버리지(Code Coverage)

코드 커버리지는 시스템 또는 SW가 테스트 케이스(Test Case)에 의해 테스트된 정도를 퍼센트(%)로 표현한 것이며, 테스트의 충분함(thoroughness)을 측정하는 기법으로 공식적(구조적) 커버리지와 비공식적 커버리지가 존재한다.

공식적(구조적) 커버리지는 의사가 환자를 돌볼 때 겉으로 보이는 증상 뿐 만 아니라 어떤 상황에서는 환자의 내부에 대한 다양한 검사(혈액형검사 또는 CT, MRI 등)까지도 고려해야 문제를 일찍 발견 하듯이 내부의 제어 흐름, 구조 등을 테스트 하는 것으로 그림과 같이 6가지(구문, 결정, 조건, 조건/결정, 변형조건/결정, 다중조건) 종류가 있다.



[그림 2 커버리지 종류]



[그림 3 소나큐브에서 제공하는 커버리지]

## 1.2 복잡도(Complexity)

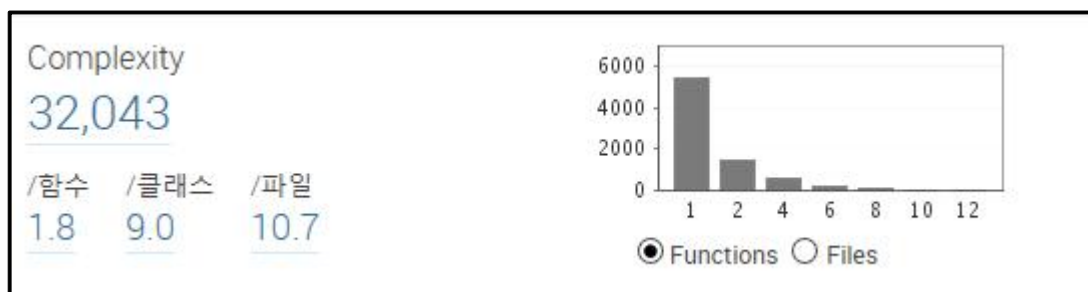
복잡도는 문제의 복잡도, 프로그램 구조의 복잡도, 자료의 복잡도 등 다양하게 존재하며 여기에서는 프로그램 구조(논리)의 복잡도를 다룬다.

프로그램 구조 복잡도는 시스템을 어느 정도까지 시험해야 하는지 또는 시스템을 개발하는데 어느 정도의 노력을 들여야 하는지를 결정하는 근거를 제공한다.

측정한 결과 복잡도가 높은 경우, 일반적으로 많은 장애가 유발될 가능성이 크므로 보다 정밀한 시험을 통해 내재된 오류를 사전에 제거하여야 한다.

일반적으로 복잡도는 시스템 구성 요소의 크기를 제어하는 논리의 복잡도 조합으로 볼 수 있다.

코드가 복잡한 정도를 정량적으로 수치화하기 위해 일반적으로 도구를 사용하고 있으며, 복잡도 중에 대표적으로 활용되는 복잡도는 if, else, switch 등의 조건문이 많은 코드의 복잡한 정도를 수치화한 사이클로매틱 복잡도(Cyclomatic complexity)이다.



[그림 4 소나큐브에서 제공하는 복잡도]

## 1.3 중복도(Duplication Code)

중복도는 개발자가 가장 흔히 저지르는 실수로, 중복 코드가 많게 되면, 코드를 수정할 때 중복된 모든 부분을 수행해야 하므로 유지보수의 어려움이 발생한다. 특히, 여러 개의 자료가 하나의 사실에 대해서 나타내고 있다면, 그 내용을 모두 같게 하여 일관성을 유지해주어야 하며, 갱신(Update)이 있을 경우에도 같은 내용을 갖고 있는 자료들을 모두 갱신해야하기 때문에 경제성이 떨어질 수 있다. 따라서 중복성이 높은 곳은 소스코드 리팩토링을 통해 잠재적 결함 발생 가능성을 낮추어야 한다.



[그림 5 소나큐브의 코드 중복]

## 1.4 정적 분석(Static Analysis)

SW의 소스코드와 모델에서 결함을 발견하는 것으로 SW코드를 수행하는 동적 테스트에 비해 정적 분석은 조사대상 SW를 실제로 실행하지 않는 상태에서 도구의 지원으로 수행한다.

### □ 정적 분석의 특징

- 테스트 실행 전에 조기에 결함을 발견
- 장애(Failures)보다는 결함(Defects)을 발견
- 코드와 설계의 유지보수성(Maintainability) 향상 지원
- 코드와 설계의 복잡도(Complexity) 및 중복도(Duplication) 측정
- 결함 예방 기능

### □ 정적 분석의 단점

- 경고 메시지 관리 (False Positive)
- 정의된 패턴을 벗어나는 임의적 결함은 찾지 못함 (소스코드 기반)
- 상용 도구의 경우 고가이지만 다양한 오픈소스가 존재함

### □ 정적 분석을 통해 다음과 같은 결함을 제거할 수 있다.

(도구의 종류와 언어에 따라 제약)

- 정의되지 않은 값으로 변수 참조
- 모듈과 컴포넌트 간에 일관되지 않은 인터페이스
- 사용되지 않는 변수
- 사용되지 않는 코드 (dead code)
- 코딩 표준(스타일) 위반
- 보안 취약성
- 코드와 SW 모델의 구문 규칙(syntax) 위반

## 2 측정 및 분석 절차

### 2.1 측정 및 분석 계획 수립

측정이란 어떤 사물에 대한 특성을 정해진 단위를 기준으로 재는 활동으로, 정성적이 아닌 정량적인 데이터를 확보하도록 함으로써 보다 객관적인 현상 파악과 의사결정을 가능하게 하는 중요한 활동이다.

일반적인 SW 개발 활동에 있어서의 측정 대상은 개발 규모(Size), 투입 자원(Resource), 시간(Time), 품질(Quality)과 관계있는 측정치를 선정한다.

측정 및 분석 계획은 측정할 대상을 선정하고 선정된 측정치를 언제, 어떻게, 누가 수집하고, 어떻게 활용할 것인가를 정의하는 것으로 다음과 같은 활동을 포함한다.

- 1) 프로젝트 특성을 고려한 측정 및 분석의 목적, 절차, 기준 등을 정의
- 2) 측정 및 분석 목적에 부합하는 측정항목을 선정하되, 프로젝트의 일정, 비용, 공수, 품질 관련된 항목을 포함함. 특히, 프로젝트 초반에 제품품질목표(사업범위 내 정량적 목표, 핵심 기능목표, 주요 비기능 목표), 프로세스 품질목표(일정준수율 등), 코드품질관련 목표를 구체적으로 설정
- 3) 측정 및 분석 방법 및 시기, 분석 결과의 활용 방법 등을 포함한 측정 및 분석 활동 계획을 수립
- 4) 측정 및 분석 계획을 이해관계자들과 합의하고 공유

### 2.2 측정 및 분석 실시

측정 및 분석 계획에 따라 정해진 시점에 해당 측정 항목 데이터를 수집하도록 한다. 데이터의 수집은 일반적으로 문서화된 양식을 활용하고, 소스코드의 경우 자동화된 툴을 통해 이루어지게 되며, 수집된 데이터는 정해진 데이터 저장소에 보관되어야 하며 올바른 데이터가 수집되기 위해서는 무엇보다도 일관된 프로세스와 일관된 기준의 측정치 정의가 선행되어야 한다.

일관된 프로세스가 정의되지 않은 상태에서 수집된 데이터는 서로 다른 환경 하에서 발생한 데이터이므로 비교를 통해 나온 결과를 분석에 활용할 수 없기 때문이다. 또한 기준이 다른 측정치가 수집되었을 경우에도 데이터의 비교를 통해 나온 결과를 개선 시 적용하기 어려울뿐더러 의사소통의 어려움이 발생할 수 있

기 때문이다. 프로젝트 관리 차원의 데이터 분석은 범위, 일정, 자원, 품질의 관점에서 현재의 프로젝트 상태를 파악하여 프로젝트 진행상 현재 문제점은 없는지, 위험 발생의 가능성은 없는지를 파악하는 것이다.

데이터 분석에는 다양한 방법 (Pie Chart, Pareto Diagram, Histogram, Fishbone Diagram, Control Chart) 등이 활용 된다.

## 2.3 측정 및 분석 결과 관리

측정 및 분석 활동의 결과를 관리하는 것은 수집된 원본 데이터와 분석 결과 데이터를 프로젝트에서 사전에 정의된 저장소에 저장하는 것은 물론 프로젝트 종료 후 조직의 자산으로써 사전에 정해진 저장소에 저장하여 관리한다.

### 3 측정지표

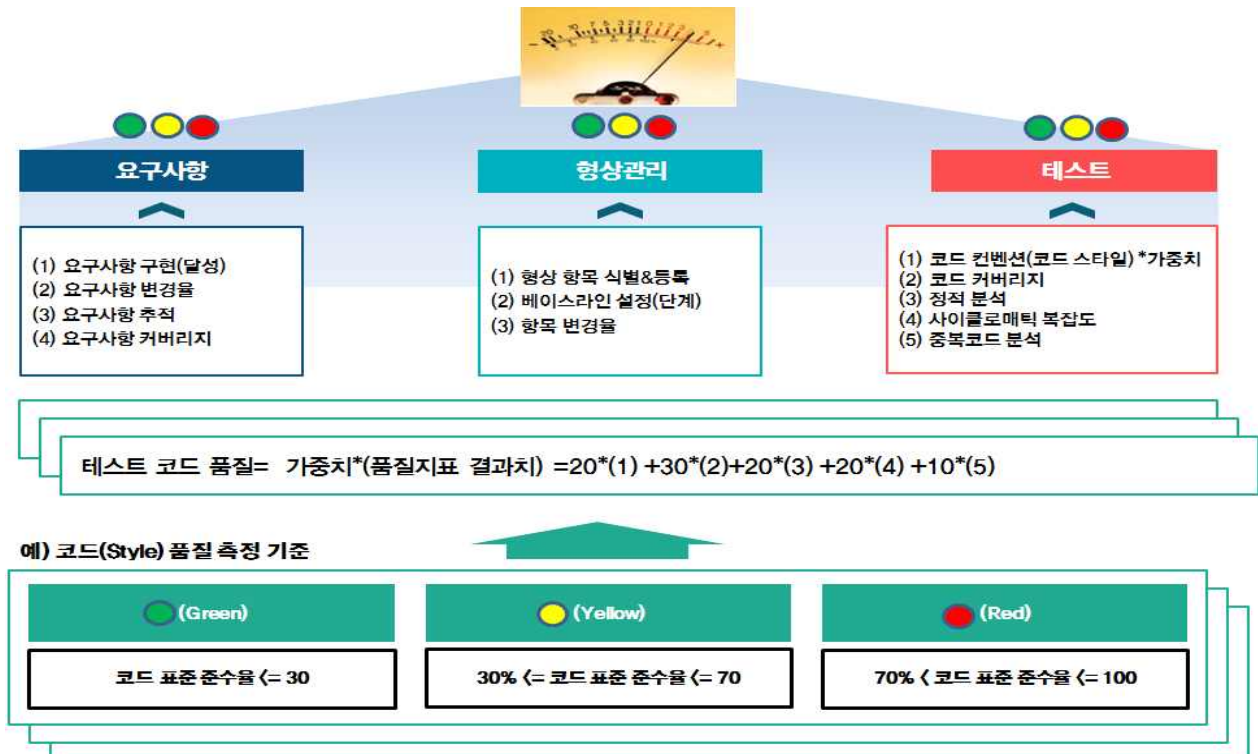
측정 활동에서 중요한 것은 무엇을 측정할 지를 정의하는 것으로 프로젝트에서 꼭 필요한 데이터만을 측정하는 것이 중요하다.

실제 측정 및 분석 활동은 시간과 비용을 많이 소요하는 활동이기 때문에 많은 측정치를 선정하여 측정하는 경우 측정 활동의 실패로 연결될 가능성이 높다. 또한 LOC(Line Of Code)등 단일 지표만으로는 한계가 있어 복잡도, 중복도, 커버리지, LOC 등을 활용하여 여러 지표를 함께 분석하는 것이 위험을 최소화할 수 있다.

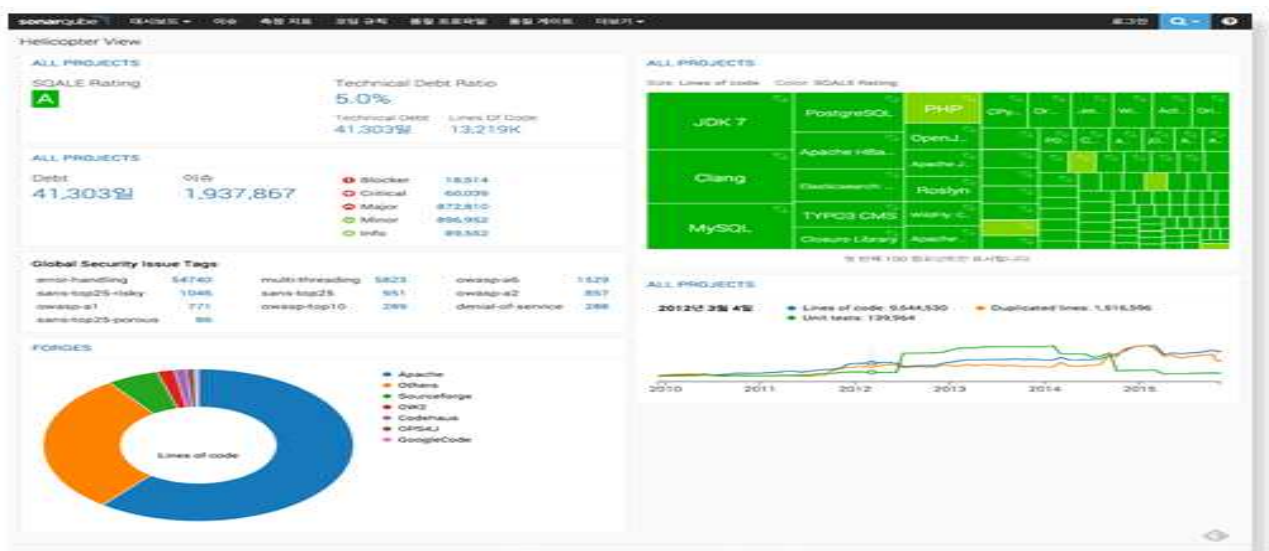


### 3.1 품질 대시보드(Quality Dashboard)

품질 대시보드는 프로젝트 품질에 대해 여러 각도로 분석하여 개선방향을 도출하거나 개선 효과를 검증하는 근거로 활용할 수 있는 데이터를 축적하고 프로젝트 품질 지표를 관리하여 가시성을 확보하는 것이 목적이다.



[ 그림 6 품질 대시보드 예시 ]



[ 그림 7 소나큐브 코드 품질 대시보드 예시 ]

## 3.2 품질 지표 사례

구분	품질지표	평가방법	품질 목표	합격기준 허용편차범위
요구 사항	요구사항 달성율	□ 측정수식 $C = (A \div B) \times 100$ A : 구현(반영)된 요구사항의 수(단계누적) B : 정의(승인)된 요구사항의 수	100%	- 90% 이상 : 정상 - 90% 미만 : 개선
	요구사항 변경율	□ 측정수식 $C = (A \div B) \times 100$ A : 당월에 변경(추가, 수정, 삭제)된 요구사 항 수 B : (전월) 요구사항의 수	20% 이하	- 20% 이하 : 정상 - 20% 이상 : 개선
	요구사항 추적율	□ 측정수식 $= A / B \times 100$ A : 순방향 추적 가능 요구사항 수 B : 기능 요구사항 수	100%	
	요구사항 커버리지	□ 측정수식 $C = (A \div B) \times 100$ A : 테스트케이스 수 B : 요구사항 별	100%	- 95% 이상 : 정상 - 95% 이하 : 개선
테스팅	결함제거율	□ 측정수식 $C = (A \div B) \times 100$ A : 결함 해결 건수 B : 결함발생 건수	100%	95%이상 (심각도, 중요도)
	테스트 수행율 (단위)	□ 측정수식 $C = (A \div B) \times 100$ A : 실행된 테스트케이스 수 (누적) B : 계획된 총 테스트 케이스 수	95% 이상	- 95% 이상 : 정상 - 95% 이하 : 개선
	테스트 수행율(통합)	□ 측정수식 $C = (A \div B) \times 100$ A : 실행된 테스트케이스 수 (누적) B : 계획된 총 테스트 케이스 수	95% 이상	- 95% 이상 : 정상 - 95% 이하 : 개선
	테스트 수행율 (시스템)	□ 측정수식 $C = (A \div B) \times 100$ A : 실행된 테스트케이스 수 (누적) B : 계획된 총 테스트 케이스 수	95% 이상	- 95% 이상 : 정상 - 95% 이하 : 개선
	테스트 합격율	□ 측정수식 $C = (A \div B) \times 100$ A : Pass된 건수 B : 총 테스트 수행 건수		
	결함밀도	□ 측정수식 $C = (A \div B) \times 100$ A : 발생 결함 B : KLOC OR FP	0.05/KLOC 이하	

코드 품질 5대 지표	코드 커버리지	<input type="checkbox"/> 측정수식 $\text{Statement} = A/B \times 100$ A.테스트 수행된 스테이트먼트 수(개) B. 스테이트먼트 수(개)	70%	$\pm 15\%$
		<input type="checkbox"/> 측정수식 $\text{branch} = A/B \times 100$ A.테스트 수행된 브랜치 수(개) B. 브랜치 수(개)	50%	$\pm 15\%$
	코딩표준 준수율	<input type="checkbox"/> 측정수식 $\text{표준 준수율} = A/B \times 100$ A. 위배된 위배 수 B. 코딩표준 수	100%	$\pm 5\%$
	정적 분석 결함제거율	<input type="checkbox"/> 측정수식 $= A/B \times 100$ A. 제거된 결함 수 B. Rule을 위배한 결함 수	100%	$\pm 5\%$
	사이클로매틱 복잡도	사이클로매틱 측정	25 이상	10%이내



# SW코드품질 측정과 분석

---

발행일 : 2015년 11월

편 집 : 정보통신산업진흥원 소프트웨어공학센터

제 작 : 정보통신산업진흥원

가 격 : 비매품

---

- \* 본 자료는 GCS(Global Creative Software) 수행기관을 대상으로 SW공학 관련 교육을 제공하는 GCS School의 교육 교재를 요약한 내용입니다.
- \* 본 책자 내용의 무단 전재를 금하며, 가공 및 인용 시에는 반드시 출처를 밝혀주시기 바랍니다.
- \* 본 책자에 대한 문의사항은 정보통신산업진흥원 소프트웨어공학센터로 연락바랍니다.  
(02-2132-1347, 소프트웨어공학센터 SW품질혁신팀)