Final Report

System Test, Static Analysis Test

SW V&V, CTIP, … (ANT, SVN, Mantis)

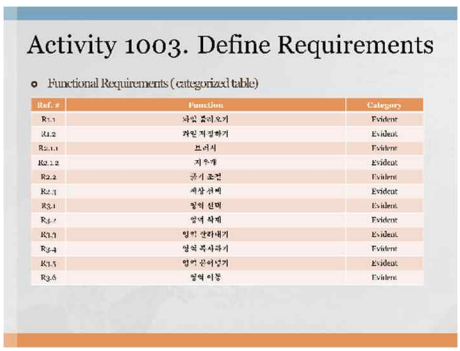
Software Verification T2

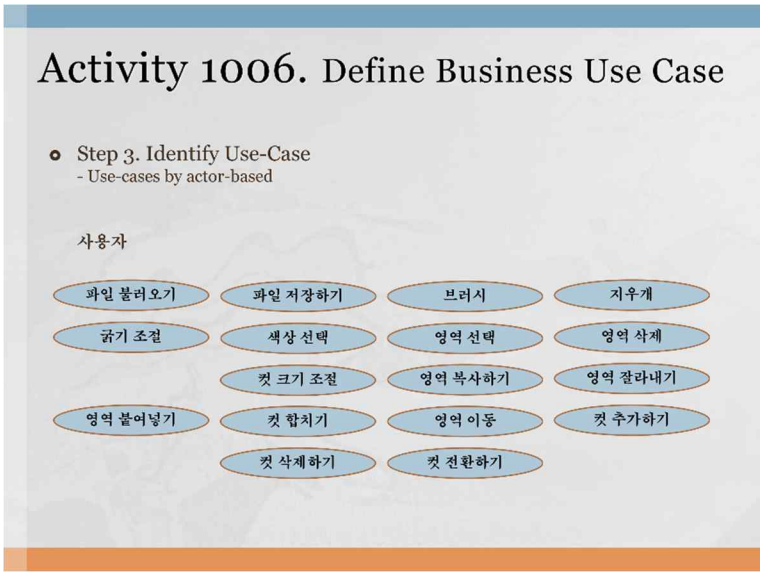
1. Category-Partition, Pairwise Testing

1.1 Testable Features

OSP1000ver4_team2
OSP2030ver3_team2
OSP2040ver2_team2

업데이트 된 3 가지 문서를 기준으로 다시 Category-Partition Testing 을 수행하였습니다. Testable Features 에는 아래와 같은 것들이 있습니다.





Testable Features			
파일 불러오기	브러시	영역 선택하기	컷 추가하기
파일 저장하기	지우개	영역 삭제하기	컷 삭제하기
	~~굵기 조절하기~~	영역 잘라내기	컷 크기조절하기
	~~색상 선택하기~~	영역 복사하기	컷 합치기
		영역 붙여넣기	컷 전환하기
		영역 이동하기	

1.2 파일 불러오기

1.2.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
파일 불러오기	파일 이름 및 경로
	파일 타입
	이미지 파일의 Width
	이미지 파일의 Height
	컷 사이즈

1.2.2 Step 2: Identify Representative Values

Input Parameters	Test Values
파일 이름 및 경로	empty
	존재하는 파일 이름 및 경로
	존재하지 않는 파일 이름 및 경로
파일 타입	bmp
	png
	others
이미지 파일의 Width	0
	100
	others
이미지 파일의 Height	0
	100
	others
컷 사이즈	0
	1
	others

1.2.3 Step 3: Generate Test Case Specifications

1.2.3.1 Error Constraints

Input Parameters	Test Values	Constraints
파일 이름 및 경로	empty	[error]
	존재하지 않는 파일 이름 및 경로	[error]
파일 타입	others	[error]
이미지 파일의 Width	0	[error]
이미지 파일의 Height	0	[error]

1.2.3.2 Property Constraints

Input Parameters	Test Values	Constraints
파일 이름 및 경로	존재하는 파일 이름 및 경로	[property RQFN]
파일 타입	bmp	[property RQFT]
	png	[property RQFT]
이미지 파일의 Width	0	[if RQFN, RQFT][error]
	100	[if RQFN, RQFT]
	others	[if RQFN, RQFT]
이미지 파일의 Height	0	[if RQFN, RQFT][error]
	100	[if RQFN, RQFT]
	others	[if RQFN, RQFT]

1.2.3.3 Single Constraints

1.2.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	24 Test Cases
Step 3 : Property Constraints	24 Test Cases
Step 3 : Single Constraints	

1.2.4 Pairwise

file_name	file_type	image_width	image_height	total_cut_size
exist	bmp	100	100	0
	png	others	others	1
				others

1.2.4.1 Test Cases

case	file_type	image_width	image_height	total_cut_size	pairings	pass/fail
1	bmp	100	100	0	6	pass
2	png	others	others	0	6	pass
3	bmp	others	100	1	5	pass

4	png	100	others	1	5	pass
5	bmp	100	others	others	4	pass
6	png	others	100	others	4	pass

1.2.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
total_cut_size	file_type	0	bmp	1	1
total_cut_size	file_type	0	png	1	2
total_cut_size	file_type	1	bmp	1	3
total_cut_size	file_type	1	png	1	4
total_cut_size	file_type	others	bmp	1	5
total_cut_size	file_type	others	png	1	6
total_cut_size	image_width	0	100	1	1
total_cut_size	image_width	0	others	1	2
total_cut_size	image_width	1	100	1	4
total_cut_size	image_width	1	others	1	3
total_cut_size	image_width	others	100	1	5
total_cut_size	image_width	others	others	1	6
total_cut_size	image_height	0	100	1	1
total_cut_size	image_height	0	others	1	2
total_cut_size	image_height	1	100	1	3
total_cut_size	image_height	1	others	1	4
total_cut_size	image_height	others	100	1	6
total_cut_size	image_height	others	others	1	5
file_type	image_width	bmp	100	2	1, 5
file_type	image_width	bmp	others	1	3
file_type	image_width	png	100	1	4
file_type	image_width	png	others	2	2, 6
file_type	image_height	bmp	100	2	1, 3
file_type	image_height	bmp	others	1	5
file_type	image_height	png	100	1	6
file_type	image_height	png	others	2	2, 4
image_width	image_height	100	100	1	1
image_width	image_height	100	others	2	4, 5
image_width	image_height	others	100	2	3, 6
image_width	image_height	others	others	1	2

1.2.4.3 Summary of Pairwise

Category-Partiton :	24 Test Cases
Pairwise :	6 Test Cases

1.3 파일 저장하기

1.3.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
파일 저장하기	파일 이름 및 경로
	파일 타입
	이미지 파일의 Width
	이미지 파일의 Height
	컷 사이즈

1.3.2 Step 2: Identify Representative Values

Input Parameters	Test Values
파일 이름 및 경로	empty
	존재하는 파일 이름 및 경로
	존재하지 않는 파일 이름 및 경로
파일 타입	png
	others
이미지 파일의 Width	0
	100
	others
이미지 파일의 Height	0
	100
	others
컷 사이즈	0
	1
	others

1.3.3 Step 3: Generate Test Case Specifications

1.3.3.1 Error Constraints

Input Parameters	Test Values	Constraints
파일 이름 및 경로	empty	[error]
	존재하는 파일 이름 및 경로	[error]
파일 타입	others	[error]
이미지 파일의 Width	0	[error]
이미지 파일의 Height	0	[error]
컷 사이즈	0	[error]

1.3.3.2 Property Constraints

Input Parameters	Test Values	Constraints
파일 이름 및 경로	존재하지 않는 파일 이름 및 경로	[property RQFN]
파일 타입	png	[property RQFT]
이미지 파일의 Width	0	[if RQFN, RQFT][error]
	100	[if RQFN, RQFT]
	others	[if RQFN, RQFT]
이미지 파일의 Height	0	[if RQFN, RQFT][error]
	100	[if RQFN, RQFT]
	others	[if RQFN, RQFT]

1.3.3.3 Single Constraints

1.3.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	8 Test Cases
Step 3 : Property Constraints	8 Test Cases
Step 3 : Single Constraints	

1.3.4 Pairwise

file_name	file_type	image_width	image_height	total_cut_size
non-exist	png	100	100	1
		others	others	others

1.3.4.1 Test Cases

case	image_width	image_height	total_cut_size	pairings	pass/fail
1	100	100	1	3	pass
2	100	others	others	3	pass
3	others	100	others	3	pass
4	others	others	1	3	pass

1.3.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
image_width	image_height	100	100	1	1
image_width	image_height	100	others	1	2
image_width	image_height	others	100	1	3
image_width	image_height	others	others	1	4
image_width	total_cut_size	100	1	1	1
image_width	total_cut_size	100	others	1	2
image_width	total_cut_size	others	1	1	4
image_width	total_cut_size	others	others	1	3
image_height	total_cut_size	100	1	1	1
image_height	total_cut_size	100	others	1	3
image_height	total_cut_size	others	1	1	4
image_height	total_cut_size	others	others	1	2

1.3.4.3 Summary of Pairwise

Category-Partiton :	8 Test Cases
Pairwise :	4 Test Cases

1.4 브러시

1.4.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
브러시	굵기 선택하기
	색상 선택하기
	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.4.2 Step 2: Identify Representative Values

Input Parameters	Test Values
굵기 선택하기	0
	1
	others
색상 선택하기	16 colors
	others
시작 X 좌표	0
	1
	out of bound
시작 Y 좌표	0
	1
	out of bound
끝 X 좌표	0
	1
	out of bound
끝 Y 좌표	0
	1
	out of bound
현재 컷 번호	0
	1

	others

1.4.3 Step 3: Generate Test Case Specifications

1.4.3.1 Error Constraints

Input Parameters	Test Values	Constraints
굵기 선택하기	others	[error]
색상 선택하기	others	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.4.3.2 Property Constraints

Input Parameters	Test Values	Constraints
굵기 선택하기	0	[property RQLS]
	1	[property RQLS]
	others	[property RQLS][error]
색상 선택하기	16 colors	[property RQCS]
	others	[property RQCS][error]
시작 X 좌표	0	[if RQLS,RQCS]
	1	[if RQLS,RQCS]
	out of bound	[if RQLS,RQCS] [error]
시작 Y 좌표	0	[if RQLS,RQCS]
	1	[if RQLS,RQCS]
	out of bound	[if RQLS,RQCS] [error]
끝 X 좌표	0	[if RQLS,RQCS]
	1	[if RQLS,RQCS]
	out of bound	[if RQLS,RQCS] [error]
끝 Y 좌표	0	[if RQLS,RQCS]
	1	[if RQLS,RQCS]
	out of bound	[if RQLS,RQCS] [error]

1.4.3.3 Single Constraints

1.4.3.4 Summary of Categories

Step 2 :	1,458 Test Cases
Step 3 : Error Constraints	64 Test Cases
Step 3 : Property Constraints	64 Test Cases
Step 3 : Single Constraints	

1.4.4 Pairwise

thickness	color	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	16 colors	0	0	0	0	1
1		1	1	1	1	others

1.4.4.1 Test Cases

case	thickness	color	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	16 colors	0	0	0	0	1	21	pass
2	1	16 colors	1	1	1	1	others	21	pass
3	0	~16 colors	0	1	0	1	others	9	pass
4	1	~16 colors	1	0	1	0	1	9	pass
5	0	~16 colors	1	1	0	0	1	4	pass
6	1	~16 colors	0	0	1	1	others	4	pass
7	0	~16 colors	~1	~0	1	0	others	2	pass
8	1	~16 colors	~0	~1	0	1	1	2	pass

1.4.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
thickness	cur_cut_num	0	1	2	1, 5
thickness	cur_cut_num	0	others	2	3, 7
thickness	cur_cut_num	1	1	2	4, 8
thickness	cur_cut_num	1	others	2	2, 6
thickness	start_x_location	0	0	2	1, 3
thickness	start_x_location	0	1	2	5, 7

thickness	start_x_location	1	0	2	6, 8
thickness	start_x_location	1	1	2	2, 4
thickness	start_y_location	0	0	2	1, 7
thickness	start_y_location	0	1	2	3, 5
thickness	start_y_location	1	0	2	4, 6
thickness	start_y_location	1	1	2	2, 8
thickness	end_x_location	0	0	3	1, 3, 5
thickness	end_x_location	0	1	1	7
thickness	end_x_location	1	0	1	8
thickness	end_x_location	1	1	3	2, 4, 6
thickness	end_y_location	0	0	3	1, 5, 7
thickness	end_y_location	0	1	1	3
thickness	end_y_location	1	0	1	4
thickness	end_y_location	1	1	3	2, 6, 8
thickness	color	0	16 colors	4	1, 3, 5, 7
thickness	color	1	16 colors	4	2, 4, 6, 8
cur_cut_num	start_x_location	1	0	2	1, 8
cur_cut_num	start_x_location	1	1	2	4, 5
cur_cut_num	start_x_location	others	0	2	3, 6
cur_cut_num	start_x_location	others	1	2	2, 7
cur_cut_num	start_y_location	1	0	2	1, 4
cur_cut_num	start_y_location	1	1	2	5, 8
cur_cut_num	start_y_location	others	0	2	6, 7
cur_cut_num	start_y_location	others	1	2	2, 3
cur_cut_num	end_x_location	1	0	3	1, 5, 8
cur_cut_num	end_x_location	1	1	1	4
cur_cut_num	end_x_location	others	0	1	3
cur_cut_num	end_x_location	others	1	3	2, 6, 7
cur_cut_num	end_y_location	1	0	3	1, 4, 5
cur_cut_num	end_y_location	1	1	1	8
cur_cut_num	end_y_location	others	0	1	7
cur_cut_num	end_y_location	others	1	3	2, 3, 6
cur_cut_num	color	1	16 colors	4	1, 4, 5, 8
cur_cut_num	color	others	16 colors	4	2, 3, 6, 7
start_x_location	start_y_location	0	0	2	1, 6
start_x_location	start_y_location	0	1	2	3, 8
start_x_location	start_y_location	1	0	2	4, 7
start_x_location	start_y_location	1	1	2	2, 5
start_x_location	end_x_location	0	0	3	1, 3, 8
start_x_location	end_x_location	0	1	1	6

start_x_location	end_x_location	1	0	1	5
start_x_location	end_x_location	1	1	3	2, 4, 7
start_x_location	end_y_location	0	0	1	1
start_x_location	end_y_location	0	1	3	3, 6, 8
start_x_location	end_y_location	1	0	3	4, 5, 7
start_x_location	end_y_location	1	1	1	2
start_x_location	color	0	16 colors	4	1, 3, 6, 8
start_x_location	color	1	16 colors	4	2, 4, 5, 7
start_y_location	end_x_location	0	0	1	1
start_y_location	end_x_location	0	1	3	4, 6, 7
start_y_location	end_x_location	1	0	3	3, 5, 8
start_y_location	end_x_location	1	1	1	2
start_y_location	end_y_location	0	0	3	1, 4, 7
start_y_location	end_y_location	0	1	1	6
start_y_location	end_y_location	1	0	1	5
start_y_location	end_y_location	1	1	3	2, 3, 8
start_y_location	color	0	16 colors	4	1, 4, 6, 7
start_y_location	color	1	16 colors	4	2, 3, 5, 8
end_x_location	end_y_location	0	0	2	1, 5
end_x_location	end_y_location	0	1	2	3, 8
end_x_location	end_y_location	1	0	2	4, 7
end_x_location	end_y_location	1	1	2	2, 6
end_x_location	color	0	16 colors	4	1, 3, 5, 8
end_x_location	color	1	16 colors	4	2, 4, 6, 7
end_y_location	color	0	16 colors	4	1, 4, 5, 7
end_y_location	color	1	16 colors	4	2, 3, 6, 8

1.4.4.3 Summary of Pairwise

Category-Partiton :	64 Test Cases
Pairwise :	8 Test Cases

1.5 지우개

1.5.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
지우개	굵기 선택하기
	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.5.2 Step 2: Identify Representative Values

Input Parameters	Test Values
굵기 선택하기	0
	1
	others
시작 X 좌표	0
	1
	out of bound
시작 Y 좌표	0
	1
	out of bound
끝 X 좌표	0
	1
	out of bound
끝 Y 좌표	0
	1
	out of bound
현재 컷 번호	0
	1
	others

1.5.3 Step 3: Generate Test Case Specifications

1.5.3.1 Error Constraints

Input Parameters	Test Values	Constraints
굵기 선택하기	others	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.5.3.2 Property Constraints

Input Parameters	Test Values	Constraints
굵기 선택하기	0	[property RQLS]
	1	[property RQLS]
	others	[property RQLS][error]
시작 X 좌표	0	[if RQLS]
	1	[if RQLS]
	out of bound	[if RQLS] [error]
시작 Y 좌표	0	[if RQLS]
	1	[if RQLS]
	out of bound	[if RQLS] [error]
끝 X 좌표	0	[if RQLS]
	1	[if RQLS]
	out of bound	[if RQLS] [error]
끝 Y 좌표	0	[if RQLS]
	1	[if RQLS]
	out of bound	[if RQLS] [error]

1.5.3.3 Single Constraints

1.5.3.4 Summary of Categories

Step 2 :	729 Test Cases
Step 3 : Error Constraints	64 Test Cases
Step 3 : Property Constraints	64 Test Cases
Step 3 : Single Constraints	

1.5.4 Pairwise

thickness	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	0	0	0	0	1
1	1	1	1	1	others

1.5.4.1 Test Cases

case	thickness	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	0	0	0	0	1	15	pass
2	0	1	1	1	1	others	15	pass
3	1	0	1	0	1	1	11	pass
4	1	1	0	1	0	others	11	pass
5	~0	0	0	1	1	1	3	pass
6	~0	1	1	0	0	others	3	pass
7	~1	0	~1	~1	~0	others	1	pass
8	~1	1	~0	~0	~1	1	1	pass

1.5.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
thickness	start_x_location	0	0	2	1, 5
thickness	start_x_location	0	1	2	2, 6
thickness	start_x_location	1	0	2	3, 7
thickness	start_x_location	1	1	2	4, 8
thickness	start_y_location	0	0	2	1, 5
thickness	start_y_location	0	1	2	2, 6

thickness	start_y_location	1	0	2	4, 8
thickness	start_y_location	1	1	2	3, 7
thickness	end_x_location	0	0	2	1, 6
thickness	end_x_location	0	1	2	2, 5
thickness	end_x_location	1	0	2	3, 8
thickness	end_x_location	1	1	2	4, 7
thickness	end_y_location	0	0	2	1, 6
thickness	end_y_location	0	1	2	2, 5
thickness	end_y_location	1	0	2	4, 7
thickness	end_y_location	1	1	2	3, 8
thickness	cur_cut_num	0	1	2	1, 5
thickness	cur_cut_num	0	others	2	2, 6
thickness	cur_cut_num	1	1	2	3, 8
thickness	cur_cut_num	1	others	2	4, 7
start_x_location	start_y_location	0	0	2	1, 5
start_x_location	start_y_location	0	1	2	3, 7
start_x_location	start_y_location	1	0	2	4, 8
start_x_location	start_y_location	1	1	2	2, 6
start_x_location	end_x_location	0	0	2	1, 3
start_x_location	end_x_location	0	1	2	5, 7
start_x_location	end_x_location	1	0	2	6, 8
start_x_location	end_x_location	1	1	2	2, 4
start_x_location	end_y_location	0	0	2	1, 7
start_x_location	end_y_location	0	1	2	3, 5
start_x_location	end_y_location	1	0	2	4, 6
start_x_location	end_y_location	1	1	2	2, 8
start_x_location	cur_cut_num	0	1	3	1, 3, 5
start_x_location	cur_cut_num	0	others	1	7
start_x_location	cur_cut_num	1	1	1	8
start_x_location	cur_cut_num	1	others	3	2, 4, 6
start_y_location	end_x_location	0	0	2	1, 8
start_y_location	end_x_location	0	1	2	4, 5
start_y_location	end_x_location	1	0	2	3, 6
start_y_location	end_x_location	1	1	2	2, 7
start_y_location	end_y_location	0	0	2	1, 4
start_y_location	end_y_location	0	1	2	5, 8
start_y_location	end_y_location	1	0	2	6, 7
start_y_location	end_y_location	1	1	2	2, 3
start_y_location	cur_cut_num	0	1	3	1, 5, 8
start_y_location	cur_cut_num	0	others	1	4

start_y_location	cur_cut_num	1	1	1	3
start_y_location	cur_cut_num	1	others	3	2, 6, 7
end_x_location	end_y_location	0	0	2	1, 6
end_x_location	end_y_location	0	1	2	3, 8
end_x_location	end_y_location	1	0	2	4, 7
end_x_location	end_y_location	1	1	2	2, 5
end_x_location	cur_cut_num	0	1	3	1, 3, 8
end_x_location	cur_cut_num	0	others	1	6
end_x_location	cur_cut_num	1	1	1	5
end_x_location	cur_cut_num	1	others	3	2, 4, 7
end_y_location	cur_cut_num	0	1	1	1
end_y_location	cur_cut_num	0	others	3	4, 6, 7
end_y_location	cur_cut_num	1	1	3	3, 5, 8
end_y_location	cur_cut_num	1	others	1	2

1.5.4.3 Summary of Pairwise

Category-Partiton :	64 Test Cases
Pairwise :	8 Test Cases

1.6 영역 선택하기

1.6.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 선택하기	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.6.2 Step 2: Identify Representative Values

Input Parameters	Test Values
시작 X 좌표	0
	50
	out of bound
시작 Y 좌표	0
	50
	out of bound
끝 X 좌표	0
	100
	out of bound
끝 Y 좌표	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.6.3 Step 3: Generate Test Case Specifications

1.6.3.1 Error Constraints

Input Parameters	Test Values	Constraints
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.6.3.2 Property Constraints

Input Parameters	Test Values	Constraints
시작 X 좌표	0	[property RQXL]
	50	[property RQXL]
	out of bound	[property RQXL][error]
시작 Y 좌표	0	[property RQYL]
	50	[property RQYL]
	out of bound	[property RQYL][error]
끝 X 좌표	0	[if RQXL, RQYL]
	100	[if RQXL, RQYL]
	out of bound	[if RQXL, RQYL] [error]
끝 Y 좌표	0	[if RQXL, RQYL]
	100	[if RQXL, RQYL]
	out of bound	[if RQXL, RQYL] [error]

1.6.3.3 Single Constraints

1.6.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	32 Test Cases
Step 3 : Property Constraints	32 Test Cases
Step 3 : Single Constraints	

1.6.4 Pairwise

start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
0	0	0	0	1
50	50	100	100	others

1.6.4.1 Test Cases

case	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	0	0	0	1	10	pass
2	0	50	100	100	others	10	pass
3	50	0	100	0	others	8	pass
4	50	50	0	100	1	8	pass
5	~0	0	0	100	others	2	pass
6	~0	50	100	0	1	2	pass

1.6.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
start_x_location	start_y_location	0	0	2	1, 5
start_x_location	start_y_location	0	50	2	2, 6
start_x_location	start_y_location	50	0	1	3
start_x_location	start_y_location	50	50	1	4
start_x_location	end_x_location	0	0	2	1, 5
start_x_location	end_x_location	0	100	2	2, 6
start_x_location	end_x_location	50	0	1	4
start_x_location	end_x_location	50	100	1	3
start_x_location	end_y_location	0	0	2	1, 6
start_x_location	end_y_location	0	100	2	2, 5
start_x_location	end_y_location	50	0	1	3
start_x_location	end_y_location	50	100	1	4
start_x_location	cur_cut_num	0	1	2	1, 6
start_x_location	cur_cut_num	0	others	2	2, 5
start_x_location	cur_cut_num	50	1	1	4
start_x_location	cur_cut_num	50	others	1	3
start_y_location	end_x_location	0	0	2	1, 5
start_y_location	end_x_location	0	100	1	3
start_y_location	end_x_location	50	0	1	4
start_y_location	end_x_location	50	100	2	2, 6
start_y_location	end_y_location	0	0	2	1, 3
start_y_location	end_y_location	0	100	1	5
start_y_location	end_y_location	50	0	1	6
start_y_location	end_y_location	50	100	2	2, 4
start_y_location	cur_cut_num	0	1	1	1

start_y_location	cur_cut_num	0	others	2	3, 5
start_y_location	cur_cut_num	50	1	2	4, 6
start_y_location	cur_cut_num	50	others	1	2
end_x_location	end_y_location	0	0	1	1
end_x_location	end_y_location	0	100	2	4, 5
end_x_location	end_y_location	100	0	2	3, 6
end_x_location	end_y_location	100	100	1	2
end_x_location	cur_cut_num	0	1	2	1, 4
end_x_location	cur_cut_num	0	others	1	5
end_x_location	cur_cut_num	100	1	1	6
end_x_location	cur_cut_num	100	others	2	2, 3
end_y_location	cur_cut_num	0	1	2	1, 6
end_y_location	cur_cut_num	0	others	1	3
end_y_location	cur_cut_num	100	1	1	4
end_y_location	cur_cut_num	100	others	2	2, 5

1.6.4.3 Summary of Pairwise

Category-Partiton :	32 Test Cases
Pairwise :	6 Test Cases

1.7 영역 삭제하기

1.7.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 삭제하기	영역 선택 유무
	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.7.2 Step 2: Identify Representative Values

Input Parameters	Test Values
영역 선택 유무	true
	false
시작 X 좌표	0
	50
	out of bound
시작 Y 좌표	0
	50
	out of bound
끝 X 좌표	0
	100
	out of bound
끝 Y 좌표	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.7.3 Step 3: Generate Test Case Specifications

1.7.3.1 Error Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	false	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.7.3.2 Property Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	true	[property ISSL]
	false	[property ISSL][error]
시작 X 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
시작 Y 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
끝 X 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]
끝 Y 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]

1.7.3.3 Single Constraints

1.7.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	32 Test Cases
Step 3 : Property Constraints	32 Test Cases
Step 3 : Single Constraints	

1.7.4 Pairwise

isSelected	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
true	0	0	0	0	1
	50	50	100	100	others

1.7.4.1 Test Cases

case	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	0	0	0	1	10	pass
2	0	50	100	100	others	10	pass
3	50	0	100	0	others	8	pass
4	50	50	0	100	1	8	pass
5	~0	0	0	100	others	2	pass
6	~0	50	100	0	1	2	pass

1.7.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
start_x_location	start_y_location	0	0	2	1, 5
start_x_location	start_y_location	0	50	2	2, 6
start_x_location	start_y_location	50	0	1	3
start_x_location	start_y_location	50	50	1	4
start_x_location	end_x_location	0	0	2	1, 5
start_x_location	end_x_location	0	100	2	2, 6
start_x_location	end_x_location	50	0	1	4
start_x_location	end_x_location	50	100	1	3
start_x_location	end_y_location	0	0	2	1, 6
start_x_location	end_y_location	0	100	2	2, 5
start_x_location	end_y_location	50	0	1	3
start_x_location	end_y_location	50	100	1	4
start_x_location	cur_cut_num	0	1	2	1, 6
start_x_location	cur_cut_num	0	others	2	2, 5
start_x_location	cur_cut_num	50	1	1	4
start_x_location	cur_cut_num	50	others	1	3
start_y_location	end_x_location	0	0	2	1, 5
start_y_location	end_x_location	0	100	1	3
start_y_location	end_x_location	50	0	1	4
start_y_location	end_x_location	50	100	2	2, 6
start_y_location	end_y_location	0	0	2	1, 3
start_y_location	end_y_location	0	100	1	5
start_y_location	end_y_location	50	0	1	6
start_y_location	end_y_location	50	100	2	2, 4
start_y_location	cur_cut_num	0	1	1	1

start_y_location	cur_cut_num	0	others	2	3, 5
start_y_location	cur_cut_num	50	1	2	4, 6
start_y_location	cur_cut_num	50	others	1	2
end_x_location	end_y_location	0	0	1	1
end_x_location	end_y_location	0	100	2	4, 5
end_x_location	end_y_location	100	0	2	3, 6
end_x_location	end_y_location	100	100	1	2
end_x_location	cur_cut_num	0	1	2	1, 4
end_x_location	cur_cut_num	0	others	1	5
end_x_location	cur_cut_num	100	1	1	6
end_x_location	cur_cut_num	100	others	2	2, 3
end_y_location	cur_cut_num	0	1	2	1, 6
end_y_location	cur_cut_num	0	others	1	3
end_y_location	cur_cut_num	100	1	1	4
end_y_location	cur_cut_num	100	others	2	2, 5

1.7.4.3 Summary of Pairwise

Category-Partiton :	32 Test Cases
Pairwise :	6 Test Cases

1.8 영역 잘라내기

1.8.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 잘라내기	영역 선택 유무
	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.8.2 Step 2: Identify Representative Values

Input Parameters	Test Values
영역 선택 유무	true
	false
시작 X 좌표	0
	50
	out of bound
시작 Y 좌표	0
	50
	out of bound
끝 X 좌표	0
	100
	out of bound
끝 Y 좌표	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.8.3 Step 3: Generate Test Case Specifications

1.8.3.1 Error Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	false	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.8.3.2 Property Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	true	[property ISSL]
	false	[property ISSL][error]
시작 X 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
시작 Y 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
끝 X 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]
끝 Y 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]

1.8.3.3 Single Constraints

1.8.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	32 Test Cases
Step 3 : Property Constraints	32 Test Cases
Step 3 : Single Constraints	

1.8.4 Pairwise

isSelected	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
true	0	0	0	0	1
	50	50	100	100	others

1.8.4.1 Test Cases

case	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	0	0	0	1	10	pass
2	0	50	100	100	others	10	pass
3	50	0	100	0	others	8	pass
4	50	50	0	100	1	8	pass
5	~0	0	0	100	others	2	pass
6	~0	50	100	0	1	2	pass

1.8.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
start_x_location	start_y_location	0	0	2	1, 5
start_x_location	start_y_location	0	50	2	2, 6
start_x_location	start_y_location	50	0	1	3
start_x_location	start_y_location	50	50	1	4
start_x_location	end_x_location	0	0	2	1, 5
start_x_location	end_x_location	0	100	2	2, 6
start_x_location	end_x_location	50	0	1	4
start_x_location	end_x_location	50	100	1	3
start_x_location	end_y_location	0	0	2	1, 6
start_x_location	end_y_location	0	100	2	2, 5
start_x_location	end_y_location	50	0	1	3
start_x_location	end_y_location	50	100	1	4
start_x_location	cur_cut_num	0	1	2	1, 6
start_x_location	cur_cut_num	0	others	2	2, 5
start_x_location	cur_cut_num	50	1	1	4
start_x_location	cur_cut_num	50	others	1	3
start_y_location	end_x_location	0	0	2	1, 5
start_y_location	end_x_location	0	100	1	3
start_y_location	end_x_location	50	0	1	4
start_y_location	end_x_location	50	100	2	2, 6
start_y_location	end_y_location	0	0	2	1, 3
start_y_location	end_y_location	0	100	1	5
start_y_location	end_y_location	50	0	1	6
start_y_location	end_y_location	50	100	2	2, 4
start_y_location	cur_cut_num	0	1	1	1

start_y_location	cur_cut_num	0	others	2	3, 5
start_y_location	cur_cut_num	50	1	2	4, 6
start_y_location	cur_cut_num	50	others	1	2
end_x_location	end_y_location	0	0	1	1
end_x_location	end_y_location	0	100	2	4, 5
end_x_location	end_y_location	100	0	2	3, 6
end_x_location	end_y_location	100	100	1	2
end_x_location	cur_cut_num	0	1	2	1, 4
end_x_location	cur_cut_num	0	others	1	5
end_x_location	cur_cut_num	100	1	1	6
end_x_location	cur_cut_num	100	others	2	2, 3
end_y_location	cur_cut_num	0	1	2	1, 6
end_y_location	cur_cut_num	0	others	1	3
end_y_location	cur_cut_num	100	1	1	4
end_y_location	cur_cut_num	100	others	2	2, 5

1.8.4.3 Summary of Pairwise

Category-Partiton :	32 Test Cases
Pairwise :	6 Test Cases

1.9 영역 복사하기

1.9.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 복사하기	영역 선택 유무
	시작 X 좌표
	시작 X 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.9.2 Step 2: Identify Representative Values

Input Parameters	Test Values
영역 선택 유무	true
	false
시작 X 좌표	0
	50
	out of bound
시작 Y 좌표	0
	50
	out of bound
끝 X 좌표	0
	100
	out of bound
끝 Y 좌표	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.9.3 Step 3: Generate Test Case Specifications

1.9.3.1 Error Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	false	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	[error]
현재 컷 번호	0	[error]

1.9.3.2 Property Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	true	[property ISSL]
	false	[property ISSL][error]
시작 X 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
시작 Y 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
끝 X 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]
끝 Y 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]

1.9.3.3 Single Constraints

1.9.3.4 Summary of Categories

Step 2 :	243 Test Cases
Step 3 : Error Constraints	32 Test Cases
Step 3 : Property Constraints	32 Test Cases
Step 3 : Single Constraints	

1.9.4 Pairwise

isSelected	start_x_location	start_y_location	end_x_location	end_y_location	cur_cut_num
true	0	0	0	0	1
	50	50	100	100	others

1.9.4.1 Test Cases

case	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	0	0	0	0	1	10	pass
2	0	50	100	100	others	10	pass
3	50	0	100	0	others	8	pass
4	50	50	0	100	1	8	pass
5	~0	0	0	100	others	2	pass
6	~0	50	100	0	1	2	pass

1.9.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
start_x_location	start_y_location	0	0	2	1, 5
start_x_location	start_y_location	0	50	2	2, 6
start_x_location	start_y_location	50	0	1	3
start_x_location	start_y_location	50	50	1	4
start_x_location	end_x_location	0	0	2	1, 5
start_x_location	end_x_location	0	100	2	2, 6
start_x_location	end_x_location	50	0	1	4
start_x_location	end_x_location	50	100	1	3
start_x_location	end_y_location	0	0	2	1, 6
start_x_location	end_y_location	0	100	2	2, 5
start_x_location	end_y_location	50	0	1	3
start_x_location	end_y_location	50	100	1	4
start_x_location	cur_cut_num	0	1	2	1, 6
start_x_location	cur_cut_num	0	others	2	2, 5
start_x_location	cur_cut_num	50	1	1	4
start_x_location	cur_cut_num	50	others	1	3
start_y_location	end_x_location	0	0	2	1, 5
start_y_location	end_x_location	0	100	1	3
start_y_location	end_x_location	50	0	1	4
start_y_location	end_x_location	50	100	2	2, 6
start_y_location	end_y_location	0	0	2	1, 3
start_y_location	end_y_location	0	100	1	5
start_y_location	end_y_location	50	0	1	6
start_y_location	end_y_location	50	100	2	2, 4
start_y_location	cur_cut_num	0	1	1	1

start_y_location	cur_cut_num	0	others	2	3, 5
start_y_location	cur_cut_num	50	1	2	4, 6
start_y_location	cur_cut_num	50	others	1	2
end_x_location	end_y_location	0	0	1	1
end_x_location	end_y_location	0	100	2	4, 5
end_x_location	end_y_location	100	0	2	3, 6
end_x_location	end_y_location	100	100	1	2
end_x_location	cur_cut_num	0	1	2	1, 4
end_x_location	cur_cut_num	0	others	1	5
end_x_location	cur_cut_num	100	1	1	6
end_x_location	cur_cut_num	100	others	2	2, 3
end_y_location	cur_cut_num	0	1	2	1, 6
end_y_location	cur_cut_num	0	others	1	3
end_y_location	cur_cut_num	100	1	1	4
end_y_location	cur_cut_num	100	others	2	2, 5

1.9.4.3 Summary of Pairwise

Category-Partiton :	32 Test Cases
Pairwise :	6 Test Cases

1.10 영역 붙여넣기

1.10.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 붙여넣기	클립보드 유무
	클립보드 영역 Width
	클립보드 영역 Height
	현재 컷 번호

1.10.2 Step 2: Identify Representative Values

Input Parameters	Test Values
클립보드 유무	true
	false
클립보드 영역 Width	0
	100
	out of bound
클립보드 영역 Height	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.10.3 Step 3: Generate Test Case Specifications

1.10.3.1 Error Constraints

Input Parameters	Test Values	Constraints
클립보드 유무	false	[error]
클립보드 영역 Width	0	[error]
클립보드 영역 Height	0	[error]
현재 컷 번호	0	[error]

1.10.3.2 Property Constraints

Input Parameters	Test Values	Constraints
클립보드 유무	true	[property ISCB]
	false	[property ISCB][error]
클립보드 영역 Width	0	[if ISCB][error]
	100	[if ISCB]
	out of bound	[if ISCB]
클립보드 영역 Height	0	[if ISCB][error]
	100	[if ISCB]
	out of bound	[if ISCB]

1.10.3.3 Single Constraints

1.10.3.4 Summary of Categories

Step 2 :	54 Test Cases
Step 3 : Error Constraints	8 Test Cases
Step 3 : Property Constraints	8 Test Cases
Step 3 : Single Constraints	

1.10.4 Pairwise

isClipboard	image_width	image_height	cur_cut_num
TRUE	0	0	1
	100	100	others
	out of bound	out of bound	

1.10.4.1 Test Cases

case	image_width	image_height	cur_cut_num	pairings	pass/fail
1	100	100	1	3	pass
2	100	out of bound	others	3	pass
3	out of bound	100	others	3	pass
4	out of bound	out of bound	1	3	pass

1.10.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
image_width	image_height	100	100	1	1
image_width	image_height	100	out of bound	1	2
image_width	image_height	out of bound	100	1	3
image_width	image_height	out of bound	out of bound	1	4
image_width	cur_cut_num	100	1	1	1
image_width	cur_cut_num	100	others	1	2
image_width	cur_cut_num	out of bound	1	1	4
image_width	cur_cut_num	out of bound	others	1	3
image_height	cur_cut_num	100	1	1	1

image_height	cur_cut_num	100	others	1	3
image_height	cur_cut_num	out of bound	1	1	4
image_height	cur_cut_num	out of bound	others	1	2

1.10.4.3 Summary of Pairwise

Category-Partiton :	8 Test Cases
Pairwise :	4 Test Cases

1.11 영역 이동하기

1.11.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
영역 이동하기	영역 선택 유무
	이미지 Width
	이미지 Height
	시작 X 좌표
	시작 Y 좌표
	끝 X 좌표
	끝 Y 좌표
	현재 컷 번호

1.11.2 Step 2: Identify Representative Values

Input Parameters	Test Values
영역 선택 유무	true
	false
이미지 Width	0
	100
	out of bound
이미지 Height	0
	100
	out of bound

시작 X 좌표	0
	50
	out of bound
시작 Y 좌표	0
	50
	out of bound
끝 X 좌표	0
	100
	out of bound
끝 Y 좌표	0
	100
	out of bound
현재 컷 번호	0
	1
	others

1.11.3 Step 3: Generate Test Case Specifications

1.11.3.1 Error Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	false	[error]
이미지 Width	0	[error]
이미지 Height	0	[error]
시작 X 좌표	out of bound	[error]
시작 Y 좌표	out of bound	[error]
끝 X 좌표	out of bound	[error]
끝 Y 좌표	out of bound	
현재 컷 번호	0	[error]

1.11.3.2 Property Constraints

Input Parameters	Test Values	Constraints
영역 선택 유무	true	[property ISSL]
	false	[property ISSL][error]
이미지 Width	0	[if ISSL][error]
	100	[if ISSL]

	out of bound	[if ISSL]
이미지 Height	0	[if ISSL][error]
	100	[if ISSL]
	out of bound	[if ISSL]
시작 X 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
시작 Y 좌표	0	[if ISSL]
	50	[if ISSL]
	out of bound	[if ISSL][error]
끝 X 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]
끝 Y 좌표	0	[if ISSL]
	100	[if ISSL]
	out of bound	[if ISSL][error]

1.11.3.3 Single Constraints

1.11.3.4 Summary of Categories

Step 2 :	4374 Test Cases
Step 3 : Error Constraints	128 Test Cases
Step 3 : Property Constraints	128 Test Cases
Step 3 : Single Constraints	

1.11.4 Pairwise

isClipboard	image _width	image _height	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num
TRUE	100	100	0	0	0	0	1
	out of bound	out of bound	50	50	100	100	others

1.11.4.1 Test Cases

case	image _width	image _height	start_x _location	start_y _location	end_x _location	end_y _location	cur_cut _num	pairings	pass /fail
1	100	100	0	0	0	0	1	21	pass
2	100	out of bound	50	50	100	100	others	21	pass
3	out of bound	100	50	0	100	0	others	15	pass
4	out of bound	out of bound	0	50	0	100	1	15	pass
5	~100	100	0	50	100	0	1	4	pass
6	~100	out of bound	50	0	0	100	others	4	pass
7	~out of bound	100	50	~50	~0	100	1	2	pass
8	~out of bound	out of bound	0	~0	~100	0	others	2	pass

1.11.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
image_width	image_height	100	100	2	1, 5
image_width	image_height	100	out of bound	2	2, 6
image_width	image_height	out of bound	100	2	3, 7
image_width	image_height	out of bound	out of bound	2	4, 8
image_width	start_x_location	100	0	2	1, 5
image_width	start_x_location	100	50	2	2, 6
image_width	start_x_location	out of bound	0	2	4, 8
image_width	start_x_location	out of bound	50	2	3, 7
image_width	start_y_location	100	0	2	1, 6
image_width	start_y_location	100	50	2	2, 5
image_width	start_y_location	out of bound	0	2	3, 8
image_width	start_y_location	out of bound	50	2	4, 7
image_width	end_x_location	100	0	2	1, 6
image_width	end_x_location	100	100	2	2, 5
image_width	end_x_location	out of bound	0	2	4, 7
image_width	end_x_location	out of bound	100	2	3, 8
image_width	end_y_location	100	0	2	1, 5
image_width	end_y_location	100	100	2	2, 6
image_width	end_y_location	out of bound	0	2	3, 8

image_width	end_y_location	out of bound	100	2	4, 7
image_width	cur_cut_num	100	1	2	1, 5
image_width	cur_cut_num	100	others	2	2, 6
image_width	cur_cut_num	out of bound	1	2	4, 7
image_width	cur_cut_num	out of bound	others	2	3, 8
image_height	start_x_location	100	0	2	1, 5
image_height	start_x_location	100	50	2	3, 7
image_height	start_x_location	out of bound	0	2	4, 8
image_height	start_x_location	out of bound	50	2	2, 6
image_height	start_y_location	100	0	2	1, 3
image_height	start_y_location	100	50	2	5, 7
image_height	start_y_location	out of bound	0	2	6, 8
image_height	start_y_location	out of bound	50	2	2, 4
image_height	end_x_location	100	0	2	1, 7
image_height	end_x_location	100	100	2	3, 5
image_height	end_x_location	out of bound	0	2	4, 6
image_height	end_x_location	out of bound	100	2	2, 8
image_height	end_y_location	100	0	3	1, 3, 5
image_height	end_y_location	100	100	1	7
image_height	end_y_location	out of bound	0	1	8
image_height	end_y_location	out of bound	100	3	2, 4, 6
image_height	cur_cut_num	100	1	3	1, 5, 7
image_height	cur_cut_num	100	others	1	3
image_height	cur_cut_num	out of bound	1	1	4
image_height	cur_cut_num	out of bound	others	3	2, 6, 8
start_x_location	start_y_location	0	0	2	1, 8
start_x_location	start_y_location	0	50	2	4, 5
start_x_location	start_y_location	50	0	2	3, 6
start_x_location	start_y_location	50	50	2	2, 7
start_x_location	end_x_location	0	0	2	1, 4
start_x_location	end_x_location	0	100	2	5, 8
start_x_location	end_x_location	50	0	2	6, 7
start_x_location	end_x_location	50	100	2	2, 3
start_x_location	end_y_location	0	0	3	1, 5, 8
start_x_location	end_y_location	0	100	1	4
start_x_location	end_y_location	50	0	1	3
start_x_location	end_y_location	50	100	3	2, 6, 7
start_x_location	cur_cut_num	0	1	3	1, 4, 5
start_x_location	cur_cut_num	0	others	1	8
start_x_location	cur_cut_num	50	1	1	7

start_x_location	cur_cut_num	50	others	3	2, 3, 6
start_y_location	end_x_location	0	0	2	1, 6
start_y_location	end_x_location	0	100	2	3, 8
start_y_location	end_x_location	50	0	2	4, 7
start_y_location	end_x_location	50	100	2	2, 5
start_y_location	end_y_location	0	0	3	1, 3, 8
start_y_location	end_y_location	0	100	1	6
start_y_location	end_y_location	50	0	1	5
start_y_location	end_y_location	50	100	3	2, 4, 7
start_y_location	cur_cut_num	0	1	1	1
start_y_location	cur_cut_num	0	others	3	3, 6, 8
start_y_location	cur_cut_num	50	1	3	4, 5, 7
start_y_location	cur_cut_num	50	others	1	2
end_x_location	end_y_location	0	0	1	1
end_x_location	end_y_location	0	100	3	4, 6, 7
end_x_location	end_y_location	100	0	3	3, 5, 8
end_x_location	end_y_location	100	100	1	2
end_x_location	cur_cut_num	0	1	3	1, 4, 7
end_x_location	cur_cut_num	0	others	1	6
end_x_location	cur_cut_num	100	1	1	5
end_x_location	cur_cut_num	100	others	3	2, 3, 8
end_y_location	cur_cut_num	0	1	2	1, 5
end_y_location	cur_cut_num	0	others	2	3, 8
end_y_location	cur_cut_num	100	1	2	4, 7
end_y_location	cur_cut_num	100	others	2	2, 6

1.11.4.3 Summary of Pairwise

Category-Partiton :	128 Test Cases
Pairwise :	8 Test Cases

1.12 컷 추가하기

1.12.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
컷 추가하기	컷 사이즈
	현재 컷 번호

1.12.2 Step 2: Identify Representative Values

Input Parameters	Test Values
컷 사이즈	0
	1
	others
현재 컷 번호	0
	1
	others

1.12.3 Step 3: Generate Test Case Specifications

1.12.3.1 Error Constraints

Input Parameters	Test Values	Constraints
컷 사이즈		
현재 컷 번호		

1.12.3.2 Property Constraints

Input Parameters	Test Values	Constraints
컷 사이즈	0	
	1	
	others	
현재 컷 번호	0	
	1	
	others	

1.12.3.3 Single Constraints

1.12.3.4 Summary of Categories

Step 2 :	9 Test Cases
Step 3 : Error Constraints	9 Test Cases
Step 3 : Property Constraints	9 Test Cases
Step 3 : Single Constraints	

1.12.4 Pairwise

total_cut_size	cur_cut_num
0	0
1	1
others	others

1.12.4.1 Test Cases

case	total_cut_size	cur_cut_num	pairings	pass/fail
1	0	0	1	pass
2	0	1	1	pass
3	0	others	1	pass
4	1	0	1	pass
5	1	1	1	pass
6	1	others	1	pass
7	others	0	1	pass

8	others	1	1	pass

1.12.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
total_cut_size	cur_cut_num	0	0	1	1
total_cut_size	cur_cut_num	0	1	1	2
total_cut_size	cur_cut_num	0	others	1	3
total_cut_size	cur_cut_num	1	0	1	4
total_cut_size	cur_cut_num	1	1	1	5
total_cut_size	cur_cut_num	1	others	1	6
total_cut_size	cur_cut_num	others	0	1	7
total_cut_size	cur_cut_num	others	1	1	8
total_cut_size	cur_cut_num	others	others	1	9

1.12.4.3 Summary of Pairwise

Category-Partiton :	9 Test Cases
Pairwise :	9 Test Cases

1.13 컷 삭제하기 / 전환하기

1.13.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
컷 삭제하기 / 전환하기	컷 사이즈
	현재 컷 번호

1.13.2 Step 2: Identify Representative Values

Input Parameters	Test Values
컷 사이즈	0
	1
	others

현재 컷 번호	0
	1
	others

1.13.3 Step 3: Generate Test Case Specifications

1.13.3.1 Error Constraints

Input Parameters	Test Values	Constraints
컷 사이즈	0	[error]
현재 컷 번호	0	[error]

1.13.3.2 Property Constraints

Input Parameters	Test Values	Constraints
컷 사이즈	1	
	others	
현재 컷 번호	1	
	others	

1.13.3.3 Single Constraints

1.13.3.4 Summary of Categories

Step 2 :	9 Test Cases
Step 3 : Error Constraints	4 Test Cases
Step 3 : Property Constraints	4 Test Cases
Step 3 : Single Constraints	

1.13.4 Pairwise

total_cut_size	cur_cut_num
1	1
others	others

1.13.4.1 Test Cases

case	total_cut_size	cur_cut_num	pairings	pass/fail
1	1	1	1	pass
2	1	others	1	pass
3	others	1	1	pass
4	others	others	1	pass

1.13.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
total_cut_size	cur_cut_num	1	1	1	1
total_cut_size	cur_cut_num	1	others	1	2
total_cut_size	cur_cut_num	others	1	1	3
total_cut_size	cur_cut_num	others	others	1	4

1.13.4.3 Summary of Pairwise

Category-Partiton :	4 Test Cases
Pairwise :	4 Test Cases

1.14 컷 크기조절하기

1.14.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
컷 크기조절하기	컷 Width
	컷 Height
	조절할 컷 Width
	조절할 컷 Height
	현재 컷 번호

1.14.2 Step 2: Identify Representative Values

Input Parameters	Test Values
조절할 컷 Width	0
	small
	large
조절할 컷 Height	0
	small
	large
현재 컷 번호	0
	1
	others

1.14.3 Step 3: Generate Test Case Specifications

1.14.3.1 Error Constraints

Input Parameters	Test Values	Constraints
조절할 컷 Width		
조절할 컷 Height		
현재 컷 번호	0	[error]

1.14.3.2 Property Constraints

Input Parameters	Test Values	Constraints
조절할 컷 Width		
조절할 컷 Height		
현재 컷 번호		

1.14.3.3 Single Constraints

1.14.3.4 Summary of Categories

Step 2 :	27 Test Cases
Step 3 : Error Constraints	18 Test Cases

Step 3 : Property Constraints	18 Test Cases
Step 3 : Single Constraints	

1.14.4 Pairwise

dst_cut_width	dst_cut_height	cur_cut_num
0	0	1
small	small	others
large	large	

1.14.4.1 Test Cases

case	dst_cut_width	dst_cut_height	cur_cut_num	pairings	pass/fail
1	0	0	1	3	pass
2	0	small	others	3	pass
3	small	0	others	3	pass
4	small	small	1	3	pass
5	large	large	1	3	pass
6	large	0	others	2	pass
7	0	large	others	2	pass
8	small	large	~1	1	pass
9	large	small	~1	1	pass

1.14.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
dst_cut_width	dst_cut_height	0	0	1	1
dst_cut_width	dst_cut_height	0	small	1	2
dst_cut_width	dst_cut_height	0	large	1	7
dst_cut_width	dst_cut_height	small	0	1	3
dst_cut_width	dst_cut_height	small	small	1	4
dst_cut_width	dst_cut_height	small	large	1	8
dst_cut_width	dst_cut_height	large	0	1	6
dst_cut_width	dst_cut_height	large	small	1	9
dst_cut_width	dst_cut_height	large	large	1	5
dst_cut_width	cur_cut_num	0	1	1	1
dst_cut_width	cur_cut_num	0	others	2	2, 7
dst_cut_width	cur_cut_num	small	1	2	4, 8

dst_cut_width	cur_cut_num	small	others	1	3
dst_cut_width	cur_cut_num	large	1	2	5, 9
dst_cut_width	cur_cut_num	large	others	1	6
dst_cut_height	cur_cut_num	0	1	1	1
dst_cut_height	cur_cut_num	0	others	2	3, 6
dst_cut_height	cur_cut_num	small	1	2	4, 9
dst_cut_height	cur_cut_num	small	others	1	2
dst_cut_height	cur_cut_num	large	1	2	5, 8
dst_cut_height	cur_cut_num	large	others	1	7

1.14.4.3 Summary of Pairwise

Category-Partiton :	18 Test Cases
Pairwise :	9 Test Cases

1.15 컷 합치기

1.15.1 Step 1: Identify Independently Testable Features and Parameter Characteristics

Testable Features	Input Parameters
컷 합치기	컷 사이즈
	합쳐진 컷 Width
	합쳐진 컷 Height

1.15.2 Step 2: Identify Representative Values

Input Parameters	Test Values
합쳐진 컷 Width	0
	600
	others
합쳐진 컷 Height	0
	600
	others
컷 사이즈	0
	1

	others

1.15.3 Step 3: Generate Test Case Specifications

1.15.3.1 Error Constraints

Input Parameters	Test Values	Constraints
합쳐진 컷 Width	0	[error]
합쳐진 컷 Height	0	[error]
컷 사이즈	0	[error]

1.15.3.2 Property Constraints

Input Parameters	Test Values	Constraints
합쳐진 컷 Width		
합쳐진 컷 Height		
컷 사이즈		

1.15.3.3 Single Constraints

1.15.3.4 Summary of Categories

Step 2 :	18 Test Cases
Step 3 : Error Constraints	4 Test Cases
Step 3 : Property Constraints	4 Test Cases
Step 3 : Single Constraints	

1.15.4 Pairwise

total_cut_size	sum_image_width	sum_image_height
1	600	600
others	others	others

1.15.4.1 Test Cases

case	total_cut_size	sum_image _width	sum_image _height	pairings	pass/fail
1	1	600	600	3	pass
2	1	others	others	3	pass
3	others	600	others	3	pass
4	others	others	600	3	pass

1.15.4.2 Pairing Details

var1	var2	value1	value2	appearances	cases
total_cut_size	sum_image_width	1	600	1	1
total_cut_size	sum_image_width	1	others	1	2
total_cut_size	sum_image_width	others	600	1	3
total_cut_size	sum_image_width	others	others	1	4
total_cut_size	sum_image_height	1	600	1	1
total_cut_size	sum_image_height	1	others	1	2
total_cut_size	sum_image_height	others	600	1	4
total_cut_size	sum_image_height	others	others	1	3
sum_image_width	sum_image_height	600	600	1	1
sum_image_width	sum_image_height	600	others	1	3
sum_image_width	sum_image_height	others	600	1	4
sum_image_width	sum_image_height	others	others	1	2

1.15.4.3 Summary of Pairwise

Category-Partiton :	4 Test Cases
Pairwise :	4 Test Cases

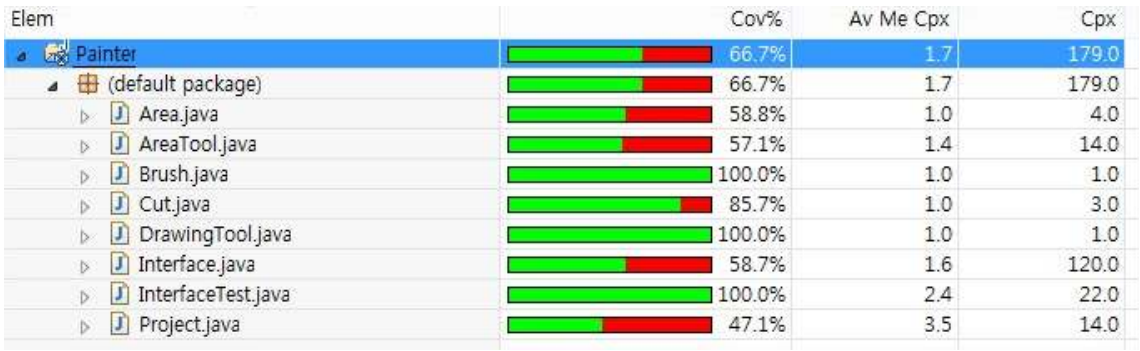
1.16 Category-Partition, Pairwise Testing 하면서...

카테고리를 뽑아낸다는 것이 사람에 따라 얼마나 달라질 수 있는지를 알았다. 같은 종류의 프로그램을 서로 다른 팀과 분석해보면서 카테고리를 나누는 레벨도 다르고, 카테고리에 사용자의 행위도 더하는 독특한 방식도 보았다. 그만큼 테스트를 수행하는 수행자가 얼마나 경험이 많고 지식이 풍부한지에 따라서 테스트는 좋은 성과를 낼 것이다. 수업 중에 테스트를 많이 한다고 해서, 프로그램의 모든 Statement 를 테스트 한다고 해서 Error 를 많이 찾아내는 것이 아니라고 배웠다. 어느 부분을 어떻게 테스트 해야 하는지 아는 것이 중요한 것 같다. Pairwise Testing 의 경우 프로그램이 Test Cases 를 뽑아내는 일이 대부분 다 해주어서 크게

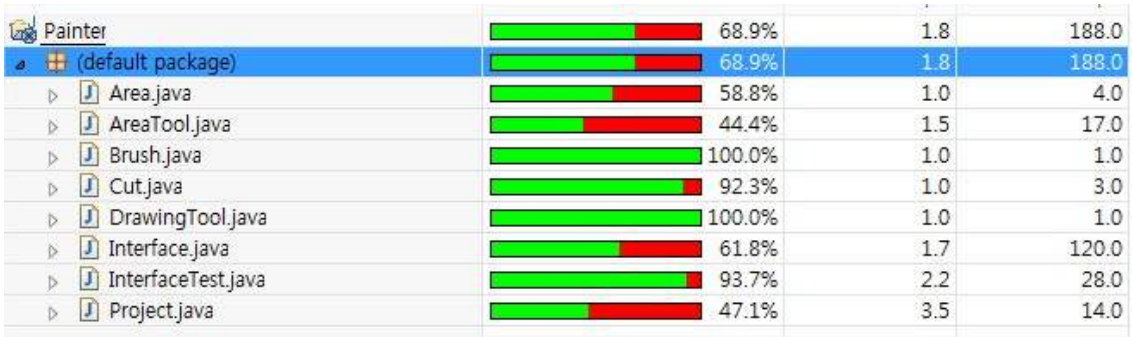
어렵지는 않았지만 놀라운 것은 Category-Partition Testing 에서 Constraints 로 열심히 줄여 놓은 Test Case 조차도 수를 줄인다는 것이다. 이보다 더 줄어들 수 없다고 생각했던 Test Cases 들도 Pairwise 는 반 이상 줄이는 것을 경험했다. 결국 Error 를 찾아냈을 때는 Test Cases 가 적은 것보다 많은 것이 Error 의 정확한 위치를 찾아내기 쉽다지만 Test Cases 를 하나, 하나 수작업으로 테스트 해봐야 하기 때문에 불필요한 Test Cases 가 줄어든다는 면에서 편리함을 느꼈다.

2. Clover

지난번과 마찬가지로 SMA Team 2 에서 자체적으로 수행한 JUnit Test 모두 성공하였다. 지난번의 경우, JUnit Test Case 가 9 개였던 것에 반해, 이번 JUnit Test Case 는 13 개로 늘었다.



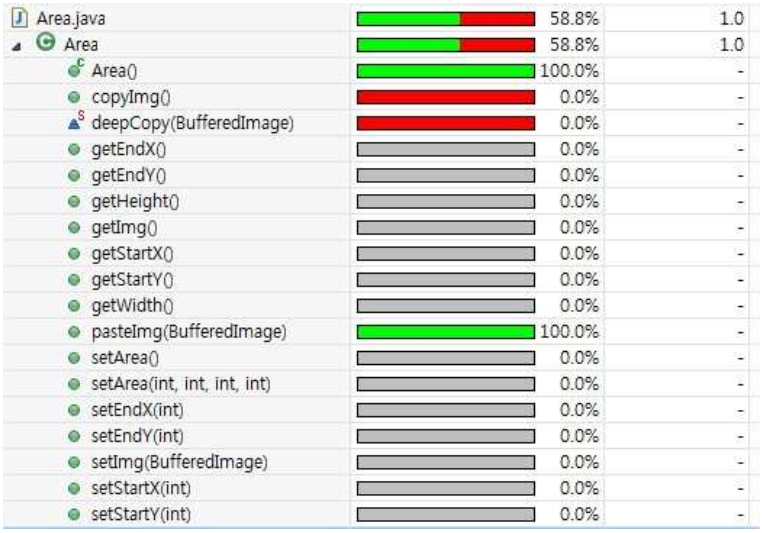
Elem	Cov%	Av Me Cpx	Cpx
Painter	66.7%	1.7	179.0
(default package)	66.7%	1.7	179.0
Area.java	58.8%	1.0	4.0
AreaTool.java	57.1%	1.4	14.0
Brush.java	100.0%	1.0	1.0
Cut.java	85.7%	1.0	3.0
DrawingTool.java	100.0%	1.0	1.0
Interface.java	58.7%	1.6	120.0
InterfaceTest.java	100.0%	2.4	22.0
Project.java	47.1%	3.5	14.0



Painter	68.9%	1.8	188.0
(default package)	68.9%	1.8	188.0
Area.java	58.8%	1.0	4.0
AreaTool.java	44.4%	1.5	17.0
Brush.java	100.0%	1.0	1.0
Cut.java	92.3%	1.0	3.0
DrawingTool.java	100.0%	1.0	1.0
Interface.java	61.8%	1.7	120.0
InterfaceTest.java	93.7%	2.2	28.0
Project.java	47.1%	3.5	14.0

Clover 수행 결과, 지난번 Code Coverage 비율이 66.7%였던 반면, 이번 JUnit Test Code 의 Code Coverage 비율은 68.9%로 70%에 근접한 Code Coverage 를 보여주었다. 많은 상승폭이 있다고 보긴 어렵지만 기존의 Code Coverage 비율이 높았던 걸 생각하면 많은 개선이 있었다고 판단된다.

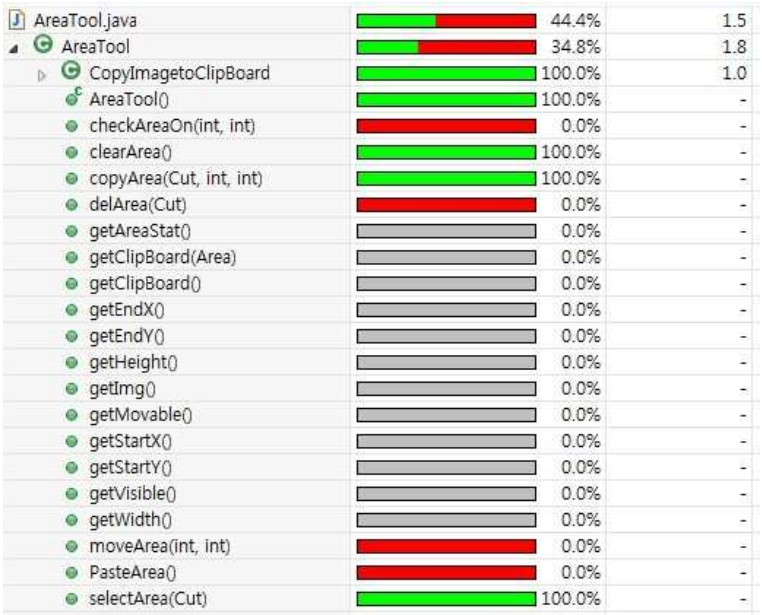
2.1 Area.java



Area.java	58.8%	1.0
Area	58.8%	1.0
Area()	100.0%	-
copyImg()	0.0%	-
deepCopy(BufferedImage)	0.0%	-
getEndX()	0.0%	-
getEndY()	0.0%	-
getHeight()	0.0%	-
getImg()	0.0%	-
getStartX()	0.0%	-
getStartY()	0.0%	-
getWidth()	0.0%	-
pasteImg(BufferedImage)	100.0%	-
setArea()	0.0%	-
setArea(int, int, int, int)	0.0%	-
setEndX(int)	0.0%	-
setEndY(int)	0.0%	-
setImg(BufferedImage)	0.0%	-
setStartX(int)	0.0%	-
setStartY(int)	0.0%	-

Area.java 의 경우, Area Class 의 Code Coverage 가 45%에서 58.8%로 상승하였는데, 지난번 Code Coverage Analysis 의 주요 이슈였던 copyImg()와 deepCopy 의 Test Case 가 생기지 않은 그대로의 상태였다. Statement Coverage 는 Code Coverage 에서 제외한 getter/setter 함수를 제외한 13 개 중 8 개였고, Branches 는 존재하지 않았다.

2.2 AreaTool.java

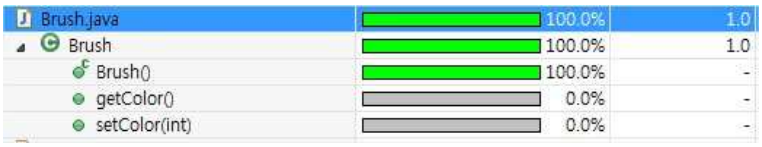


AreaTool.java	44.4%	1.5
AreaTool	34.8%	1.8
CopyImagetoClipBoard	100.0%	1.0
AreaTool()	100.0%	-
checkAreaOn(int, int)	0.0%	-
clearArea()	100.0%	-
copyArea(Cut, int, int)	100.0%	-
delArea(Cut)	0.0%	-
getAreaStat()	0.0%	-
getClipBoard(Area)	0.0%	-
getClipBoard()	0.0%	-
getEndX()	0.0%	-
getEndY()	0.0%	-
getHeight()	0.0%	-
getImg()	0.0%	-
getMovable()	0.0%	-
getStartX()	0.0%	-
getStartY()	0.0%	-
getVisible()	0.0%	-
getWidth()	0.0%	-
moveArea(int, int)	0.0%	-
PasteArea()	0.0%	-
selectArea(Cut)	100.0%	-

AreaTool.java 의 경우, AreaTool Class 의 Code Coverage 가 43.9%에서 44.4%로 증가하였는데, 지난번 분석의 주요 이슈였던 checkAreaOn()과 moveArea()에 대한 Unit Test Case 가 생성되지

않았다. 또한 추가된 주요 method 인 delArea()와 PasteArea()에 대한 Unit Test Case 가 생성되지 않았다. Code Coverage 가 상승한 이유는 오로지 Configuration 에 의한 property method 를 Coverage 에서 제외함으로써 생성된 것뿐이었다. Statement Coverage 는 전체 37 개 중 17 개였고, 이 중 Branch Coverage 는 전체 6 개중 0 개였다.

2.3 Brush.java



Brush.java 의 Brush Class 는 Code Coverage 가 100%이므로, 세부 분석은 제외하였다.

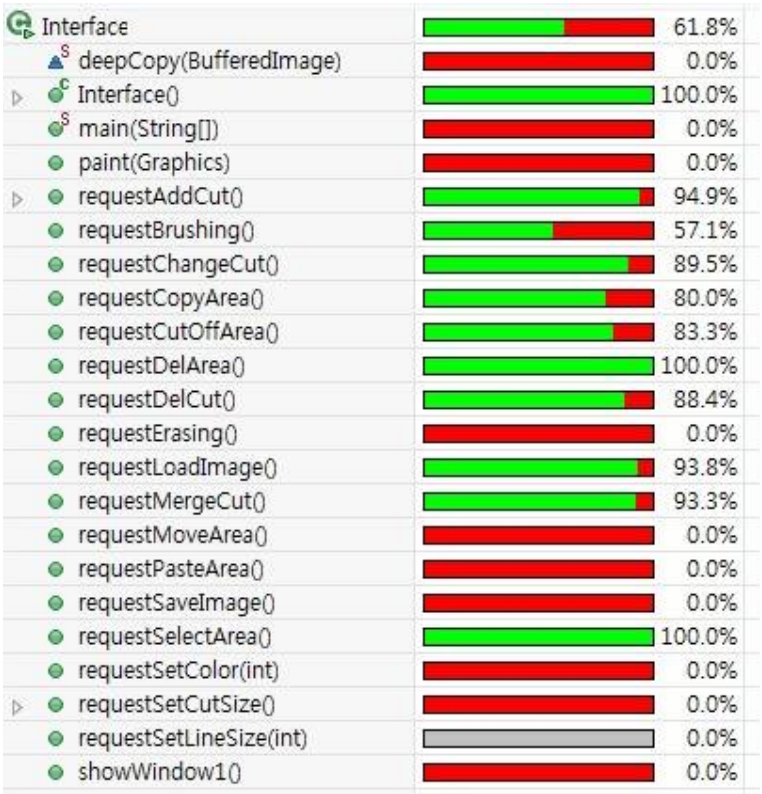
2.4 Cut.java

Cut.java 의 Cut Class 의 경우에는 Code Coverage 가 92.3%였지만 모든 method 의 Coverage 가 100%였다. 해당 클래스에서 Coverage 가 안된 Statement 는 Exception 에 관한 Statement 에 불과했으므로 이 역시 세부 분석에서 제외하였다.

2.5 DrawingTool.java

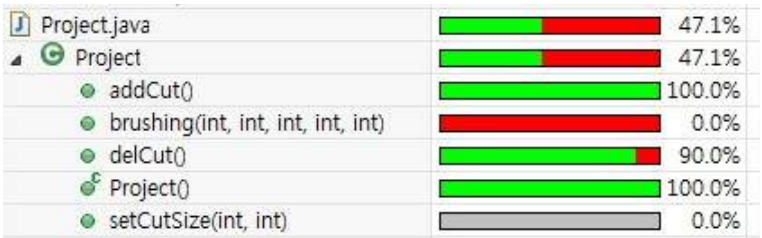
DrawingTool.java 의 Drawing Class 는 Code Coverage 가 100%이므로, 세부 분석은 제외하였다.

2.6 Interface.java



Interface.java 의 Interface Class 의 경우, 전체 69 개의 method 에서 59 개의 method 가 60% 이상의 Coverage 를 보여주었다. 지난번에 이슈가 되었던 주요 Method 에 대한 Test Case 는 전혀 만들어지지 않았지만, 기타 기존에 Unit Test 가 수행되었지만 Coverage 가 낮았던 Method 들이 Code Coverage 비율이 90%에 가깝게 상승하여, 전체 Code Coverage 역시 52.8%에서 61.8%로 증가하였다. Statement Coverage 는 전체 535 개중 293 개, Branch Coverage 는 90 개중 44 개가 Cover 가 되었다.

2.7 Project.java



마지막으로, Project Class 는 전체 4 개의 메소드 중 하나의 Method 만 Coverage 가 되지 않았는데, 이 brushing() method 가 그림판 기능의 핵심이 되는 method 여서 여전히 Test Case 가 개선되지 않은 것으로 보인다. 이전 Code Coverage 인 33.6%에서 47.1%로 증가한 이유는

Configuration 에 의한 Statement 와 Method 비율이 줄어든 것일뿐, 해당 클래스에서 Coverage 개선사항은 전혀 없다.

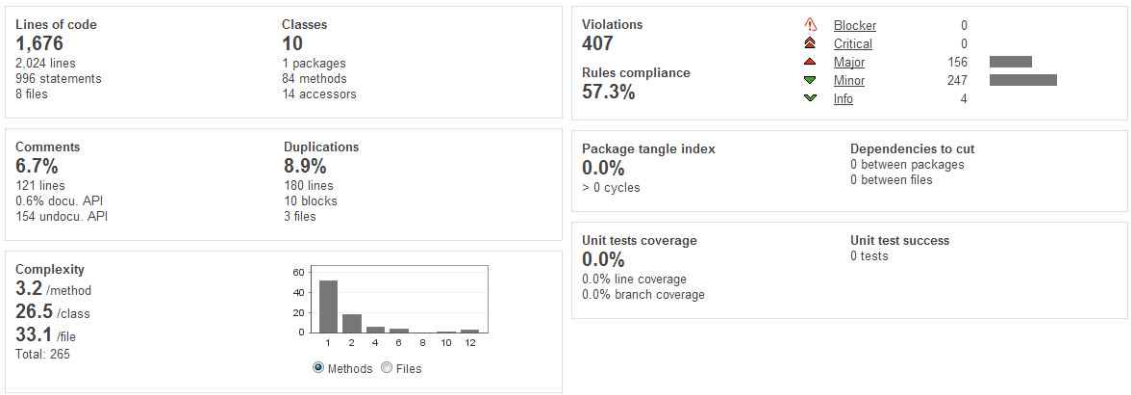
2.7 Clover Coverage Testing 하면서...

Clover 는 Code Coverage 를 측정하기에 매우 편리한 도구였다. 무엇보다 해당 Code 의 Coverage 상태를 색상 별로 표시하여 해당 부분을 수정하기 편리하도록 하였고, Code Coverage 를 보여주는 UI 도 매우 좋았다. 다만, 해당 클래스에 포함된 Method 와 Statement, Branch 의 개수를 모두 표시해주는 반면, 해당 Test Code 가 Cover 된 Statement 와 Branch 의 개수를 보여주지 않아 수치화에 어려움을 겪었던 점이 아쉬웠다.

3. Static Analysis

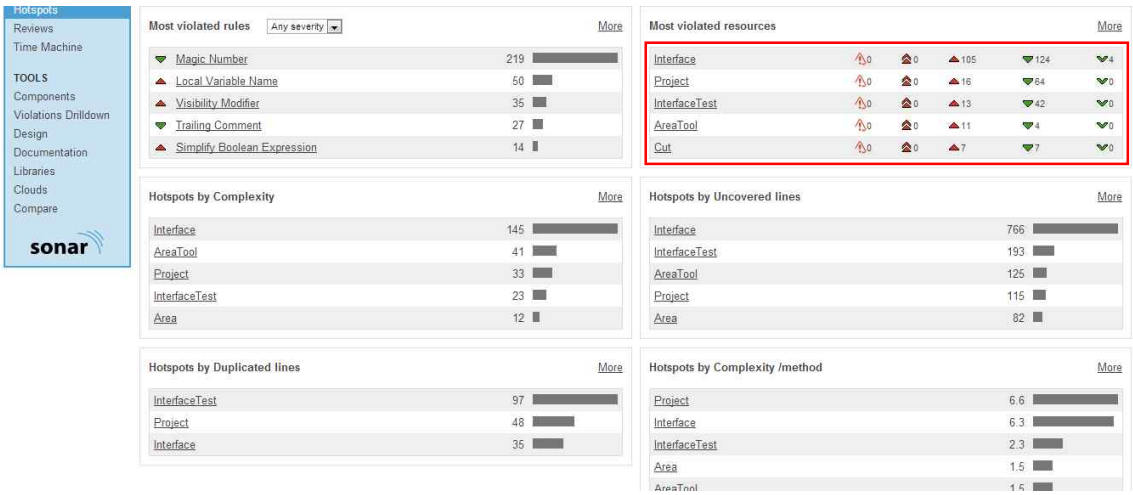
3.1 Sonar

3.1.1 Sonar Result



- Lines of code : 코드의 라인, 주요 문구, 파일 등을 보여준다.
- Classes : 패키지, 메소드, 메소드 유형을 보여준다.
- Comments : 주석의 라인 수, API 수를 보여준다.
- Duplication : 반복 라인, 블록, 파일의 수를 보여준다.
- Complexity : 메소드별, 클래스별, 파일별 Complexity 를 보여준다.
- Violations& Rules compliance : 룰에 걸리는 수치와 그에 따른 레벨별로 보여준다.
- Package tangle index : 패키지 의존성 수치와 싸이클 수를 보여준다.
- Unit test coverage : 커버리지를 나타낸다.

3.1.2.1 Painter



3.1.2.2 Violations (위반사항)

Severity (엄격한 정도)	
Blocker	0
Critical	0
Major	156
Minor	247
Info	4

Blocker 수준과 Critical 수준의 Violations 는 하나도 없기 때문에 프로그램을 잘 개발한 것으로 판단된다. 하지만 Major 수준의 Violations 가 156 개나 있다. Minor 수준의 Violations 는 정말 사소한 것들이기 때문에 생략하고, Major 수준의 Violations 에 대해 분석 결과를 다루겠다.

Rule	Count
Local Variable Name	50
Visibility Modifier	35
Simplify Boolean Expression	14
Member Name	13
Avoid Commented-Out Lines Of Code	10
If Else Stmts Must Use Braces	9
Anon Inner Length	5
If Stmts Must Use Braces	4
System Println	3
Local Final Variable Name	2
Cyclomatic Complexity	2
Method Name	2
Hidden Field	1
Naming - Suspicious Constant Field Name	1
Ncss Method Count	1
Unused Private Field	1
Unused Private Method	1
Avoid Print Stack Trace	2
Sum	156

위 그림과 같이 Major 수준의 Violations 에는 156 개가 있다.

3.1.2.3 Interface Class

Class	Interface	Count
Local Variable Name		37
Visibility Modifier		20
Simplify Boolean Expression		10
Avoid Commented-Out Lines Of Code		9
If Else Stmts Must Use Braces		4
Anon Inner Length		5
If Stmts Must Use Braces		4
Local Final Variable Name		2
Cyclomatic Complexity		1
Method Name		10
Unused Private Method		1
Avoid Print Stack Trace		2
Sum		105

Interface Class 에만 156 개의 전체 Major 수준의 Violations 중 105 개의 Major 수준의 Violations 가 있다. Local Variable Name 이라는 사소한 Rule 부터 시작해서 If 문에 중괄호 기호를 사용하라는 것 등 다양한 Rule Violations 가 있다. Sonar 에서는 이러한 Violations 를 수정하기를 권장하고 있다.

3.1.2.4 Project Class

Class	Project	Count
Visibility Modifier		4
Simplify Boolean Expression		2
Avoid Commented-Out Lines Of Code		1
If Else Stmts Must Use Braces		5
Cyclomatic Complexity		1
Method Name		2
Ncss Method Count		1
Sum		16

3.1.2.5 InterfaceTest Class

Class	InterfaceTest	Count
Local Variable Name		11
Simplify Boolean Expression		2
Sum		13

3.1.2.6 AreaTool Class

Class	AreaTool	Count
Local Variable Name		2
Visibility Modifier		2
Member Name		1
System Println		3
Hidden Field		1
Naming - Suspicious Constant Field Name		1
Unused Private Field		1
Sum		11

3.1.2.7 Cut Class

Class	Cut	Count
Visibility Modifier		5
Method Name		2
Sum		7

3.1.2.8 DrawingTool Class

Class	DrawingTool	Count
Visibility Modifier		3
Sum		3

3.1.2.9 Brush Class

Class	Brush	Count
Visibility Modifier		1
Sum		1

3.1.3 Detail Violations

3.1.3.1 About 'Name'

Member Name (13)
Local Variable Name (50)
Local Final Variable Name (2)
Naming - Suspicious Constant Field Name (1)
Method Name (2)

위 5 가지 Rule 은 변수와 메소드 이름에 관련된 것들이다. 간단하게 요약하자면 변수명이 대문자로 시작되지 않거나 하이픈(-) 또는 언더바(_)가 사용된 경우에 Sonar 는 Violations 중 하나로 보았다. Java 에서 변수명과 메소드명은 소문자로 시작하는 것을 권장하고, 클래스명은 대문자로 시작하는 것을 권장하기 때문에 하이픈(-)과 언더바(_) 사용에 대해서는 접어두더라도 대문자로 시작하는 변수명과 메소드명은 수정할 필요가 있다고 본다. 또한 Java 에서 상수처럼 사용되는 변수는 변수명을 전부 대문자로 하고, final 지시자 사용을 권장하고 있다. 'Naming - Suspicious Constant Field Name (1)'에서 'CopyImagetoClipBoard CB;'는 변수명이 전부 대문자로 되어 있어서 상수처럼 사용되는 변수로 의심된다. 하지만 final 지시자를 사용하지 않고 있다. 때문에 Sonar 에서는 final 지시자를 사용하여 상수처럼 사용되는 변수를 정의하거나 변수명을 소문자로 시작하도록 수정하는 것을 권장하고 있다.

Violations	Member Name (13)		
Line Num	Class		
46	Interface	public int SelectedTool;	Name 'SelectedTool' must match pattern '^[a-z][a-zA-Z0-9]
52	Interface	public int EnabledCutNum;	Name 'EnabledCutNum' must match pattern '^[a-z][a-zA-Z0-9]*$'.
59	Interface	public JPanel CanvasPanel;	Name 'CanvasPanel' must match pattern '^[a-z][a-zA-Z0-9]*$'.
60	Interface	public JPanel CutListPanel;	Name 'CutListPanel' must match pattern '^[a-z][a-zA-Z0-9]*$'.
61	Interface	public List<Canvas> CanvasList;	Name 'CanvasList' must match pattern '^[a-z][a-zA-Z0-9]*$'.
62	Interface	public List<JToggleButton> ButtonList;	Name 'ButtonList' must match pattern '^[a-z][a-zA-Z0-9]*$'.
63	Interface	public ButtonGroup CutListGroup = new ButtonGroup()	Name 'CutListGroup' must match pattern '^[a-z][a-zA-Z0-9]*$'.
64	Interface	public ButtonGroup ToolGroup = new ButtonGroup();	Name 'ToolGroup' must match pattern '^[a-z][a-zA-Z0-9]*$'.
65	Interface	public ButtonGroup SizeGroup = new ButtonGroup();	Name 'SizeGroup' must match pattern '^[a-z][a-zA-Z0-9]*$'.
66	Interface	public ButtonGroup ColorGroup = new ButtonGroup();	Name 'ColorGroup' must match pattern '^[a-z][a-zA-Z0-9]*$'.
26	Project	public int bef_x;	Name 'bef_x' must match pattern '^[a-z][a-zA-Z0-9]*$'.
27	Project	public int bef_y;	Name 'bef_y' must match pattern '^[a-z][a-zA-Z0-9]*$'.
32	AreaTool	CopyImagetoClipBoard CB;	Name 'CB' must match pattern '^[a-z][a-zA-Z0-9]*$'.

Violations	Local Variable Name (50)		
Line Num	Class		
570	Interface	BufferedImage CurrentImage;	Name 'CurrentImage' must match pattern '^[a-z][a-zA-Z0-9]*$'.
571	Interface	Graphics CurrentGraphic;	Name 'CurrentGraphic' must match pattern '^[a-z][a-zA-Z0-9]*$'.
639	Interface	JPanel ToolTop = new JPanel();	Name 'ToolTop' must match pattern '^[a-z][a-zA-Z0-9]*$'.
641	Interface	FlowLayout flowLayout_1 = (FlowLayout) ToolTop.getLay	Name 'flowLayout_1' must match pattern '^[a-z][a-zA-Z0-9]*$'.
647	Interface	JPanel File = new JPanel();	Name 'File' must match pattern '^[a-z][a-zA-Z0-9]*$'.
666	Interface	JButton btnNewButton_1 = new JButton("");	Name 'btnNewButton_1' must match pattern '^[a-z][a-zA-Z0-9]*$'.
681	Interface	JPanel Draw = new JPanel();	Name 'Draw' must match pattern '^[a-z][a-zA-Z0-9]*$'.
695	Interface	JToggleButton btnNewButton_3 = new JToggleButton("	Name 'btnNewButton_3' must match pattern '^[a-z][a-zA-Z0-9]*$'.
706	Interface	JPanel Area = new JPanel();	Name 'Area' must match pattern '^[a-z][a-zA-Z0-9]*$'.
721	Interface	JButton btnNewButton_5 = new JButton("");	Name 'btnNewButton_5' must match pattern '^[a-z][a-zA-Z0-9]*$'.
734	Interface	JButton btnNewButton_6 = new JButton("");	Name 'btnNewButton_6' must match pattern '^[a-z][a-zA-Z0-9]*$'.
745	Interface	JButton btnNewButton_7 = new JButton("");	Name 'btnNewButton_7' must match pattern '^[a-z][a-zA-Z0-9]*$'.
755	Interface	JButton btnNewButton_8 = new JButton("");	Name 'btnNewButton_8' must match pattern '^[a-z][a-zA-Z0-9]*$'.
771	Interface	JPanel Cut = new JPanel();	Name 'Cut' must match pattern '^[a-z][a-zA-Z0-9]*$'.
774	Interface	JButton btnNewButton_9 = new JButton("");	Name 'btnNewButton_9' must match pattern '^[a-z][a-zA-Z0-9]*$'.
785	Interface	JButton btnNewButton_10 = new JButton("");	Name 'btnNewButton_10' must match pattern '^[a-z][a-zA-Z0-9]*$'.
795	Interface	JButton btnNewButton_11 = new JButton("");	Name 'btnNewButton_11' must match pattern '^[a-z][a-zA-Z0-9]*$'.
805	Interface	JPanel ToolLeft = new JPanel();	Name 'ToolLeft' must match pattern '^[a-z][a-zA-Z0-9]*$'.
810	Interface	JPanel BrushConf = new JPanel();	Name 'BrushConf' must match pattern '^[a-z][a-zA-Z0-9]*$'.
813	Interface	JToggleButton btnNewButton_13 = new JToggleButton(	Name 'btnNewButton_13' must match pattern '^[a-z][a-zA-Z0-9]*$'.
826	Interface	JToggleButton btnNewButton_14 = new JToggleButton(	Name 'btnNewButton_14' must match pattern '^[a-z][a-zA-Z0-9]*$'.
855	Interface	JToggleButton button_1 = new JToggleButton("");	Name 'button_1' must match pattern '^[a-z][a-zA-Z0-9]*$'.
868	Interface	JToggleButton button_2 = new JToggleButton("");	Name 'button_2' must match pattern '^[a-z][a-zA-Z0-9]*$'.
881	Interface	JToggleButton button_3 = new JToggleButton("");	Name 'button_3' must match pattern '^[a-z][a-zA-Z0-9]*
894	Interface	JToggleButton button_4 = new JToggleButton("");	Name 'button_4' must match pattern '^[a-z][a-zA-Z0-9]*$'.
907	Interface	JToggleButton button_5 = new JToggleButton("");	Name 'button_5' must match pattern '^[a-z][a-zA-Z0-9]*$'.
920	Interface	JToggleButton button_6 = new JToggleButton("");	Name 'button_6' must match pattern '^[a-z][a-zA-Z0-9]*$'.
933	Interface	JToggleButton button_7 = new JToggleButton("");	Name 'button_7' must match pattern '^[a-z][a-zA-Z0-9]*$'.
946	Interface	JToggleButton button_8 = new JToggleButton("");	Name 'button_8' must match pattern '^[a-z][a-zA-Z0-9]*$'.
959	Interface	JToggleButton button_9 = new JToggleButton("");	Name 'button_9' must match pattern '^[a-z][a-zA-Z0-9]*$'.
972	Interface	JToggleButton button_10 = new JToggleButton("");	Name 'button_10' must match pattern '^[a-z][a-zA-Z0-9]*$'.
985	Interface	JToggleButton button_11 = new JToggleButton("");	Name 'button_11' must match pattern '^[a-z][a-zA-Z0-9]*$'.
998	Interface	JToggleButton button_12 = new JToggleButton("");	Name 'button_12' must match pattern '^[a-z][a-zA-Z0-9]*$'.
1011	Interface	JToggleButton button_13 = new JToggleButton("");	Name 'button_13' must match pattern '^[a-z][a-zA-Z0-9]*$'.
1024	Interface	JToggleButton button_14 = new JToggleButton("");	Name 'button_14' must match pattern '^[a-z][a-zA-Z0-9]*$'.
1037	Interface	JToggleButton button_15 = new JToggleButton("");	Name 'button_15' must match pattern '^[a-z][a-zA-Z0-9]*$'.
1086	Interface	FlowLayout flowLayout_2 = (FlowLayout) CutListPanel.ge	Name 'flowLayout_2' must match pattern '^[a-z][a-zA-Z0-9]*$'.
16	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
27	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
46	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
99	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
116	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
147	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
182	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
184	InterfaceTest	int EnabledCutNum=0;	Name 'EnabledCutNum' must match pattern '^[a-z][a-zA-Z0-9]*$'.
206	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
208	InterfaceTest	int EnabledCutNum=0;	Name 'EnabledCutNum' must match pattern '^[a-z][a-zA-Z0-9]*$'.
240	InterfaceTest	Interface IF = new Interface();	Name 'IF' must match pattern '^[a-z][a-zA-Z0-9]*$'.
85	AreaTool	Clipboard C = Toolkit.getDefaultToolkit().getSystemClipb	Name 'C' must match pattern '^[a-z][a-zA-Z0-9]*$'.
87	AreaTool	BufferedImage CI = (BufferedImage)C.getData(DataFlavo	Name 'CI' must match pattern '^[a-z][a-zA-Z0-9]*$'.

Violations	Local Final Variable Name (2)		
Line Num	Class		
684	Interface	final JToggleButton btnNewButton_2 = new JToggleBut	Name 'btnNewButton_2' must match pattern '^[a-z][a-zA-Z0-9]*$'.
710	Interface	final JToggleButton btnNewButton_4 = new JToggleBut	Name 'btnNewButton_4' must match pattern '^[a-z][a-zA-Z0-9]*$'.

Violations	Naming - Suspicious Constant Field Name (1)		
Line Num	Class		
32	AreaTool	CopyImagetoClipBoard CB;	The field name indicates a constant but its modifiers do not

Violations	Method Name (2)		
Line Num	Class		
32	Cut	public void LoadCut(String path) throws IOException	Name 'LoadCut' must match pattern '^[a-z][a-zA-Z0-9]*$'.
39	Cut	public void SaveCut(String path) throws IOException	Name 'SaveCut' must match pattern '^[a-z][a-zA-Z0-9]*$'.

3.1.3.2 About 'Visibility Modifier'

Visibility Modifier (35)

위 Rule 은 public 으로 되어 있는 변수들에 대해 private 접근지정자를 사용하도록 권장한다. Java 에서 Class 를 캡슐화하기 위해서 되도록이면 변수들에 대한 직접적인 접근을 제한하고 getter 나 setter 와 같은 Accessor Methods 를 사용해서 접근하도록 권장하고 있다.

Violations	Visibility Modifier (35)		
Line Num	Class		
44	Interface	public int size;	Variable 'size' must be private and have accessor methods.
45	Interface	public int color;	Variable 'color' must be private and have accessor methods.
46	Interface	public int SelectedTool;	Variable 'SelectedTool' must be private and have accessor methods.
52	Interface	public int EnabledCutNum;	Variable 'EnabledCutNum' must be private and have accessor method
54	Interface	public Project pj;	Variable 'pj' must be private and have accessor methods.
55	Interface	public DrawingTool dt;	Variable 'dt' must be private and have accessor methods.
56	Interface	public AreaTool at;	Variable 'at' must be private and have accessor methods.
59	Interface	public JPanel CanvasPanel;	Variable 'CanvasPanel' must be private and have accessor methods
60	Interface	public JPanel CutListPanel;	Variable 'CutListPanel' must be private and have accessor methods.
61	Interface	public List<Canvas> CanvasList;	Variable 'CanvasList' must be private and have accessor methods.
62	Interface	public List<JToggleButton> ButtonList;	Variable 'ButtonList' must be private and have accessor methods.
63	Interface	public ButtonGroup CutListGroup = new ButtonGroup();	Variable 'CutListGroup' must be private and have accessor methods.
64	Interface	public ButtonGroup ToolGroup = new ButtonGroup();	Variable 'ToolGroup' must be private and have accessor methods.
65	Interface	public ButtonGroup SizeGroup = new ButtonGroup();	Variable 'SizeGroup' must be private and have accessor methods.
66	Interface	public ButtonGroup ColorGroup = new ButtonGroup();	Variable 'ColorGroup' must be private and have accessor methods.
67	Interface	public Toolkit tk;	Variable 'tk' must be private and have accessor methods.
68	Interface	public Cursor brush;	Variable 'brush' must be private and have accessor methods.
69	Interface	public Cursor eraser;	Variable 'eraser' must be private and have accessor methods.
497	Interface	int x;	Variable 'x' must be private and have accessor methods.
497	Interface	int y;	Variable 'y' must be private and have accessor methods.
23	Project	public List<Cut> cuts;	Variable 'cuts' must be private and have accessor methods.
26	Project	public int bef_x;	Variable 'bef_x' must be private and have accessor methods.
27	Project	public int bef_y;	Variable 'bef_y' must be private and have accessor methods.
28	Project	public int thick;	Variable 'thick' must be private and have accessor methods.
32	AreaTool	CopyImagetoClipBoard CB;	Variable 'CB' must be private and have accessor methods.
177	AreaTool	Image i;	Variable 'i' must be private and have accessor methods.
25	Cut	public int cutnum;	Variable 'cutnum' must be private and have accessor methods.
26	Cut	public int width;	Variable 'width' must be private and have accessor methods.
27	Cut	public int height;	Variable 'height' must be private and have accessor methods.
28	Cut	public boolean cutstat	Variable 'cutstat' must be private and have accessor methods.
30	Cut	public BufferedImage img;	Variable 'img' must be private and have accessor methods.
16	DrawingTool	public Brush brush;	Variable 'brush' must be private and have accessor methods.
17	DrawingTool	public Brush eraser;	Variable 'eraser' must be private and have accessor methods.
18	DrawingTool	public int linesize;	Variable 'linesize' must be private and have accessor methods.
17	Brush	public int color;	Variable 'color' must be private and have accessor methods.

3.1.3.3 About 'If'

Simplify Boolean Expression (14)

If Stmts Must Use Braces (4)

If Else Stmts Must Use Braces (9)

위 3 가지 Rule 은 If 문 안에 사용되는 Boolean Expression 과 If 문을 묶어주는 중괄호({ }) 사용에 대해 다루고 있다. 'Simplify Boolean Expression (14)'의 경우에는 이미 If 문 조건식에

사용되는 변수가 Boolean 변수이면서도 또 다시 True 와 False 값을 비교하고 있다. 굳이 True 와 False 값을 비교하지 않더라도 If 문 조건식에서 Boolean 변수의 True 와 False 상태를 판단한다.

Violations	Simplify Boolean Expression (14)		
Line Num	Class		
211	Interface	if (ButtonList.get(i).isSelected() == true) {	Expression can be simplified.
282	Interface	if (SelectedTool == 3 && at.checkAreaOn(e.getX(), e.get	Expression can be simplified.
436	Interface	if (pj.cuts.get(i).cutstat == true) {	Expression can be simplified.
451	Interface	if (pj.cuts.get(i).cutstat == true) {	Expression can be simplified.
471	Interface	if (pj.cuts.get(i).cutstat == true) {	Expression can be simplified.
507	Interface	if (CanvasList.get(i).isVisible() == true)	Expression can be simplified.
568	Interface	if (c.cutstat == true) {	Expression can be simplified.
573	Interface	if (at.getAreaStat() == true) {	Expression can be simplified.
1059	Interface	if (CanvasList.isEmpty() == false) {	Expression can be simplified.
1061	Interface	if (CanvasList.get(i).isVisible() == true) {	Expression can be simplified.
132	Project	if (cuts.get(i).cutstat == true) {	Expression can be simplified.
151	Project	if (cuts.get(i).cutstat == true) {	Expression can be simplified.
194	InterfaceTest	if (IF.pj.cuts.get(i).cutstat == true) {	Expression can be simplified.
220	InterfaceTest	if (IF.pj.cuts.get(i).cutstat == true) {	Expression can be simplified.

Violations	If Stmts Must Use Braces (4)		
Line Num	Class		
507	Interface	if (CanvasList.get(i).isVisible() == true) CanvasList.get(i).se	Avoid using if statements without curly braces
532	Interface	if (pj.cuts.get(i).width > w) w = pj.cuts.get(i).width;	Avoid using if statements without curly braces
801	Interface	if (pj.cuts.size() > 1) requestMergeCut();	Avoid using if statements without curly braces
1071	Interface	if (e.getButton() == 1)	Avoid using if statements without curly braces

Violations	If Else Stmts Must Use Braces (9)		
Line Num	Class		
442	Interface	else CanvasList.get(i).setVisible(false);	Avoid using if...else statements without curly braces
475	Interface	if (i == ButtonList.size()) ButtonListget(i-1).setSelected(tru	Avoid using if...else statements without curly braces
477	Interface	else ButtonList.get(i).setSelected(true);	Avoid using if...else statements without curly braces
1077	Interface	else CanvasPanel.setCursor(new Cursor(Cursor.DEFAULT	Avoid using if...else statements without curly braces
85	Project	if (size > 1) thick = 2;	Avoid using if...else statements without curly braces
87	Project	else thick = 0;	Avoid using if...else statements without curly braces
135	Project	if (i == cuts.size()) cuts.get(i-1).cutstat = true;	Avoid using if...else statements without curly braces
137	Project	else cuts.get(i).cutstat = true;	Avoid using if...else statements without curly braces
141	Project	else cuts.clear();	Avoid using if...else statements without curly braces

3.1.3.4 About 'Unused & Comment'

Unused Private Field (1)
Unused Private Method (1)
Unused Imports (4)
Avoid Commented-Out Lines Of Code (10)

위 4 가지 Rule 은 프로그램 수행 중 사용하지 않는 변수와 메소드, Import 문과 Code 를 임시로 주석화 시켜놓은 것을 다루고 있다. 사용하지 않는 변수와 메소드, Import 문은 제거하는 것이 좋다. Code 를 임시로 주석화 시켜놓은 것들은 테스트 용도가 아니라면 역시 제거하는 것이 좋다.

Violations	Unused Private Field (1)		
Line Num	Class		
30	AreaTool	private int statonarea;	Avoid unused private fields such as 'statonarea'.

Violations	Unused Private Method (1)		
Line Num	Class		
1104	Interface	private static void addPopup(Component component, fi	Avoid unused private methods such as 'addPopup(Component,JPopu

Violations	Unused Imports (4)	
Line Num	Class	
	Interface	import java.awt.Image;
	Interface	import javax.imageio.ImageIO;
	Interface	import javax.swing.JScrollPane;
	Interface	import javax.swing.JScrollBar;

Violations	Avoid Commented-Out Lines Of Code (10)		
Line Num	Class		
133	Interface	//CanvasList.get(EnabledCutNum).setCursor(Cursor.getPr	This block of commented-out lines of code should be removed.
145	Interface	//CanvasList.get(EnabledCutNum).setCursor(Cursor.getPr	This block of commented-out lines of code should be removed.
240	Interface	//System.out.println("("+e.getX()+","+e.getY()+")");	This block of commented-out lines of code should be removed.
321	Interface	//System.out.println("released st " + at.getStartX() + " "	This block of commented-out lines of code should be removed.
342	Interface	//System.out.println("("+e.getX()+","+e.getY()+")");	This block of commented-out lines of code should be removed.
372	Interface	//System.out.println("dragg st " + at.getStartX() + " " + a	This block of commented-out lines of code should be removed.
390	Interface	//System.out.println("("+e.getX()+","+e.getY()+")");	This block of commented-out lines of code should be removed.
438	Interface	//System.out.println("EnabledCut:"+EnabledCutNum);	This block of commented-out lines of code should be removed.
596	Interface	/* switch(this.SelectedTool) {	This block of commented-out lines of code should be removed.
32	Project	//System.out.println(color);	This block of commented-out lines of code should be removed.

3.1.3.5 Others

Anon Inner Length (5)

Sonar 에서는 익명 내부 클래스의 라인 수를 최대 20 라인까지로 권장하고 있다. 아래에 있는 5 가지 익명 내부 클래스의 경우는 Sonar 가 권장하는 20 라인을 넘기기 때문에 라인 수를 줄일 것을 권장한다.

Violations	Anon Inner Length (5)		
Line Num	Class		
236	Interface	canvas.addMouseListener(new MouseListener() {	Anonymous inner class length is 101 lines (max allowed is 20).
337	Interface	canvas.addMouseMotionListener(new MouseMotionListe	Anonymous inner class length is 49 lines (max allowed is 20).
386	Interface	this.CanvasPanel.addMouseListener(new MouseListener(	Anonymous inner class length is 28 lines (max allowed is 20).
496	Interface	CanvasPanel.addMouseListener(new MouseAdapter() {	Anonymous inner class length is 27 lines (max allowed is 20).
1056	Interface	CanvasPanel.addMouseMotionListener(new MouseMotio	Anonymous inner class length is 27 lines (max allowed is 20).

Hidden Field (1)

for 문의 반복을 제어하는 변수로 i 라는 변수명을 사용하고 있다. 하지만 앞에서 Image i;라는 변수도 사용하고 있었다. 때문에 for 문에서 i 라는 변수를 사용하면서 Image i;라는 변수는 가려지게 된다. Sonar 는 같은 블록 내에서 이러한 같은 변수명의 재정의 사용을 피할 것을 권장하고 있다.

Violations	Hidden Field (1)		
Line Num	Class		
201	AreaTool	for (int i = 0; i < flavors.length; i++ {	'i' hides a field.

System Println (3)

Violations	System Println (3)		
Line Num	Class		
78	AreaTool	System.out.println("Copy Area to ClipBoard");	System.out.print is used
84	AreaTool	System.out.println("Load ClipBoard to Area");	System.out.print is used
172	AreaTool	System.out.println("Lost Clipboard Ownership");	System.out.print is used

Avoid Print Stack Trace (2)

Violations	Avoid Print Stack Trace (2)		
Line Num	Class		
657	Interface	e.printStackTrace();	Avoid printStackTrace(); use a logger call instead.
673	Interface	e.printStackTrace();	Avoid printStackTrace(); use a logger call instead.

Cyclomatic Complexity (2)

"Checks cyclomatic complexity of methods against a specified limit. The complexity is measured by the number of if, while, do, for, ?:, catch, switch, case statements, and operators && and || (plus one) in the body of a constructor, method, static initializer, or instance initializer. It is a measure of the minimum number of possible paths through the source and therefore the number of required tests. Generally 1-4 is considered good, 5-7 ok, 8-10 consider re-factoring, and 11+ re-factor now !"

Sonar 는 Cyclomatic Complexity 가 10 이하가 되도록 권장하고 있다. Cyclomatic Complexity 수치는 'if, while, do, for, ?:, catch, switch, case statements, and operators && and || (plus one) in the body of a constructor, method, static initializer, or instance initializer' 등의 수에 의해서 결정된다. 아래 두 메소드는 Sonar 가 권장하는 Cyclomatic Complexity 수치인 10 보다 높기 때문에 낮출 것을 권장하고 있다.

Violations	Cyclomatic Complexity (2)		
Line Num	Class		
289	Interface	@Override	Cyclomatic Complexity is 12 (max allowed is 10).
30	Project	public void brushing(int x, int y, int size, int color, int st	Cyclomatic Complexity is 21 (max allowed is 10).

NCSS Method Count (1)

NCSS 란 Non-Commented Source Code 의 라인 수를 의미한다. 순수 Code 의 라인 수이다.

	Ncss Method Count (1)		
Line Num	Class		
30	Project	public void brushing(int x, int y, int size, int color, int st	The method brushing() has an NCSS line count of 78

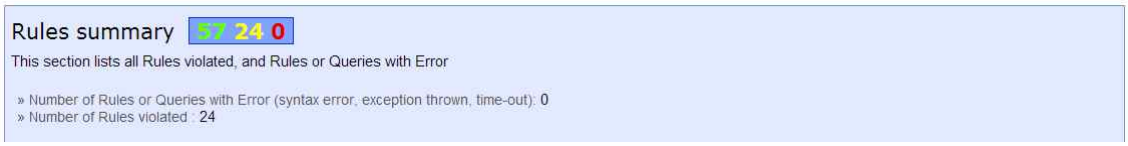
3.2 JArchitect

3.2.1 Metrics On application

해당 뷰는 프로젝트의 정보를 보여주는 곳으로 해당 프로젝트의 Code Line 개수, 즉 해당 코드가 몇 라인짜리 Code 인지를 보여주고, 그 외 프로젝트의 개수, 패키지의 개수, 메소드의 개수, 주석 라인 개수, 클래스 개수와 같은 일반 사용자가 직접 수치화하기 어려운 코드를 분석하여 보여준다.

또한 해당 뷰의 우측에서 프로젝트와 관련된 여러 수치를 확인할 수 있었는데, 해당 프로젝트의 주석 비율과 Public Method 의 비율과 같은 수치를 확인할 수 있는데, 주석 비율은 6%로 해당 프로젝트에 대한 상세한 설명이 부족했고, 대부분의 Method 가 Public 이어서 프로그램 내부 정보로 쉽게 접근할 수 있다는 점이 아쉬운 점이었다.

3.2.2 Rules Summary

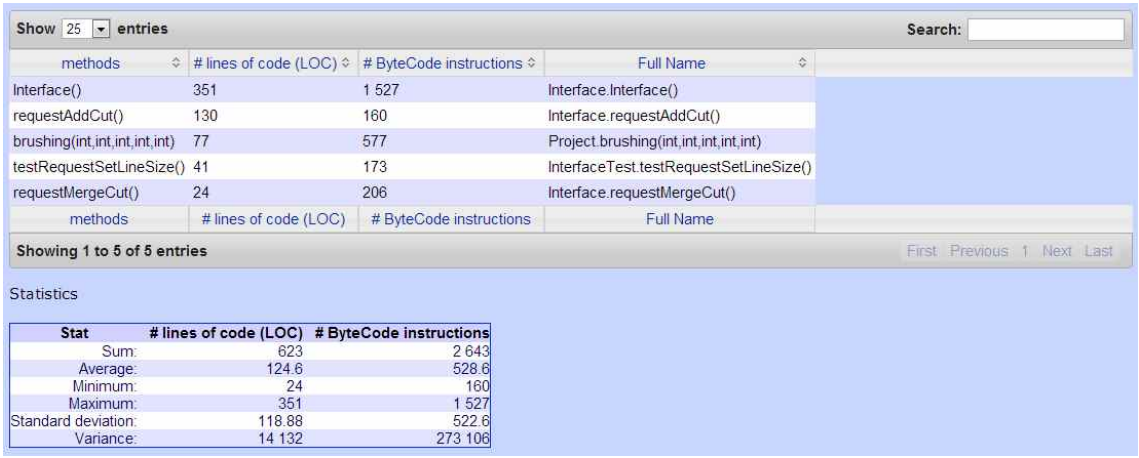


Name	# Matches	Elements	Group
Types too big - critical	1	types	Code Quality
Quick summary of methods to refactor	7	methods	Code Quality
Methods too big	5	methods	Code Quality
Methods too complex	2	methods	Code Quality
Methods potentially poorly commented	9	methods	Code Quality
Methods with too many local variables	1	methods	Code Quality
Types with too many methods	2	types	Code Quality
Types with poor cohesion	1	types	Code Quality
Class with no descendant should be final if possible	49	types	Object Oriented Design
Nested types should not be visible	2	types	Design
Instances size shouldn't be too big	1	types	Design
Always override toString	49	types	Best Practices
Potentially dead Methods	56	methods	Dead Code
Fields that could have a lower visibility	35	fields	Visibility
Fields should be declared as private	35	fields	Visibility
A field must not be assigned from outside its parent hierarchy types	8	fields	Purity - Immutability - Side-Effects
Instance fields should begin with a lower character	11	fields	Naming Conventions
Methods name should begin with an lower character	2	methods	Naming Conventions
Avoid types with name too long	1	types	Naming Conventions
Avoid methods with name too long	1	methods	Naming Conventions
Avoid defining multiple types in a source file	1	types	Source Files Organization
Name	# Matchs	Elements	Group

해당 뷰는 해당 프로젝트가 위반한 룰에 대한 정보를 표시해주는데 빨간 숫자는 Error, 노란색 숫자는 Violated , 초록색 숫자는 룰에 적합한 것을 나타낸다. 에러는 문제없이 실행이 가능했기 때문에 당연히 없었으며, Violated 자체도 메소드의 길이가 길다거나 이름이 길다거나 하는 크게 Critical 한 Violation 이 보이지 않아서 따로 SMA Team 2 에 Mantis 로 이에 대한 사항을 언급하지 않았다.

전체 프로젝트에 대한 Error, Violation 뿐만이 아니라, 해당 이슈들도 그룹별로 분류하여 보여준다. 위의 캡쳐화면과 같이 해당 이슈가 Code Quality 관련 이슈인지, 디자인 관련 이슈인지, 죽은 코드인지, 퍼포먼스에 관련된 이슈인지 등을 구분하여 보여준다. 해당 프로젝트는 각 그룹별로 다음과 같은 Violation 이 있다.

3.2.3 Methods Too Big



Show 25 entries Search:

methods	# lines of code (LOC)	# ByteCode instructions	Full Name
Interface()	351	1 527	Interface.Interface()
requestAddCut()	130	160	Interface.requestAddCut()
brushing(int,int,int,int,int)	77	577	Project.brushing(int,int,int,int,int)
testRequestSetLineSize()	41	173	InterfaceTest.testRequestSetLineSize()
requestMergeCut()	24	206	Interface.requestMergeCut()
methods	# lines of code (LOC)	# ByteCode instructions	Full Name

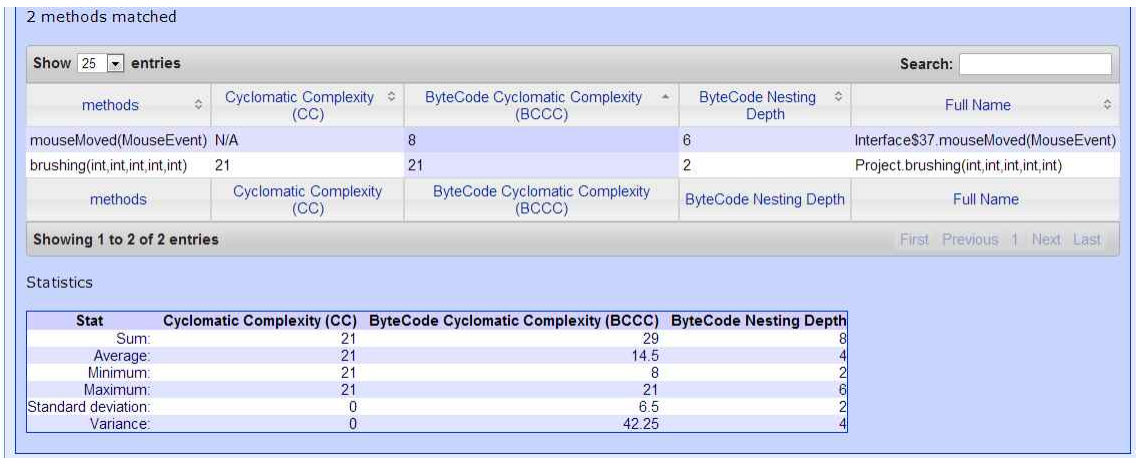
Showing 1 to 5 of 5 entries First Previous 1 Next Last

Statistics

Stat	# lines of code (LOC)	# ByteCode instructions
Sum:	623	2 643
Average:	124.6	528.6
Minimum:	24	160
Maximum:	351	1 527
Standard deviation:	118.88	522.6
Variance:	14 132	273 106

JArchitect 에서는 코드 라인이 30 줄 넘거나 바이트코드 라인이 200 줄이 넘으면 길다고 판단한다. 따라서 위에서 제시된 Method 들의 라인 수를 줄이거나 메소드 기능을 세분화하는 것을 권장하고 있다.

3.2.4 Methods Too Complex



2 methods matched

Show 25 entries Search:

methods	Cyclomatic Complexity (CC)	ByteCode Cyclomatic Complexity (BCCC)	ByteCode Nesting Depth	Full Name
mouseMoved(MouseEvent)	N/A	8	6	Interface$37.mouseMoved(MouseEvent)
brushing(int,int,int,int,int)	21	21	2	Project.brushing(int,int,int,int,int)
methods	Cyclomatic Complexity (CC)	ByteCode Cyclomatic Complexity (BCCC)	ByteCode Nesting Depth	Full Name

Showing 1 to 2 of 2 entries First Previous 1 Next Last

Statistics

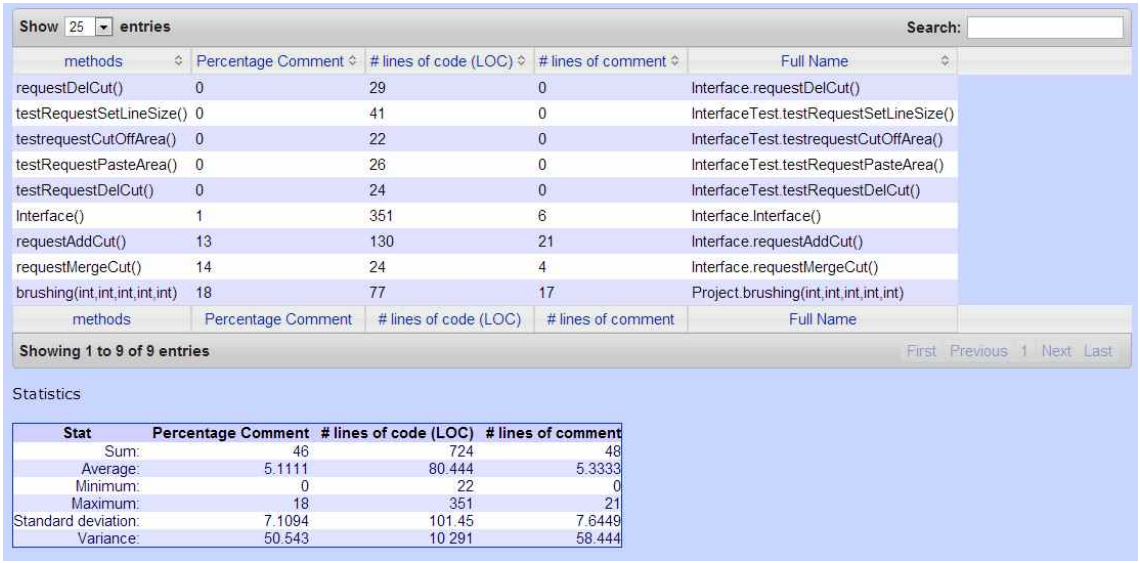
Stat	Cyclomatic Complexity (CC)	ByteCode Cyclomatic Complexity (BCCC)	ByteCode Nesting Depth
Sum:	21	29	8
Average:	21	14.5	4
Minimum:	21	8	2
Maximum:	21	21	6
Standard deviation:	0	6.5	2
Variance:	0	42.25	4

위에 제시된 해당 Method 들의 복잡도를 줄이는 것을 권장하고 있다. (참고)

CC >20 | BCCC > 40 | BCN >5 가 될 경우 복잡도가 높다고 판단하는데, 각 용어에 대한 설명은 다음과 같다.

- CC : 순환 복잡도로 if,for,while,case 같은 문들의 합
- BCCC : 점프, 분기와 같은 명령의 합

3.2.5 Methods Potentially Poorly Commented



Show 25 entries Search:

methods	Percentage Comment	# lines of code (LOC)	# lines of comment	Full Name
requestDelCut()	0	29	0	Interface.requestDelCut()
testRequestSetLineSize()	0	41	0	InterfaceTest.testRequestSetLineSize()
testrequestCutOffArea()	0	22	0	InterfaceTest.testrequestCutOffArea()
testRequestPasteArea()	0	26	0	InterfaceTest.testRequestPasteArea()
testRequestDelCut()	0	24	0	InterfaceTest.testRequestDelCut()
Interface()	1	351	6	Interface.Interface()
requestAddCut()	13	130	21	Interface.requestAddCut()
requestMergeCut()	14	24	4	Interface.requestMergeCut()
brushing(int,int,int,int,int)	18	77	17	Project.brushing(int,int,int,int,int)
methods	Percentage Comment	# lines of code (LOC)	# lines of comment	Full Name

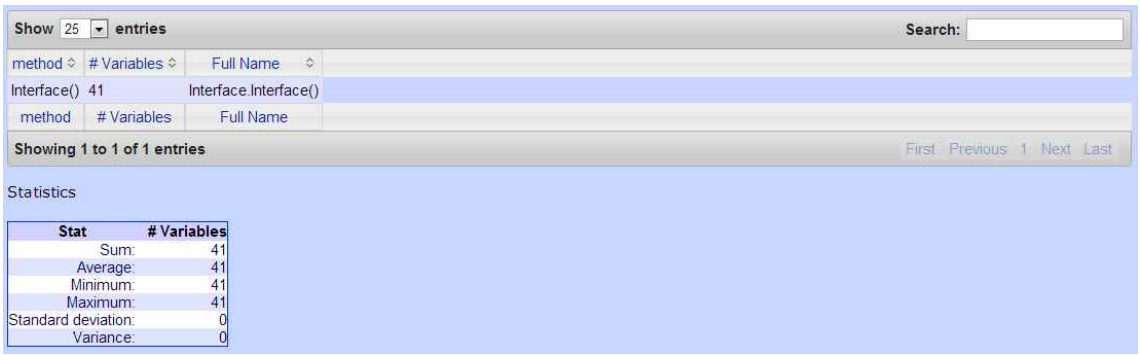
Showing 1 to 9 of 9 entries First Previous 1 Next Last

Statistics

Stat	Percentage Comment	# lines of code (LOC)	# lines of comment
Sum:	46	724	48
Average:	5.1111	80.444	5.3333
Minimum:	0	22	0
Maximum:	18	351	21
Standard deviation:	7.1094	101.45	7.6449
Variance:	50.543	10 291	58.444

해당 Method 들은 주석을 첨부하는 것을 권장하는데, 20 라인이 넘는 코드들은 주석에 의한 이해가 없을 경우 해당 Method 의 동작 방식을 이해하기 어렵기 때문이다.

3.2.6 Methods With Too Many Local Variables



Show 25 entries Search:

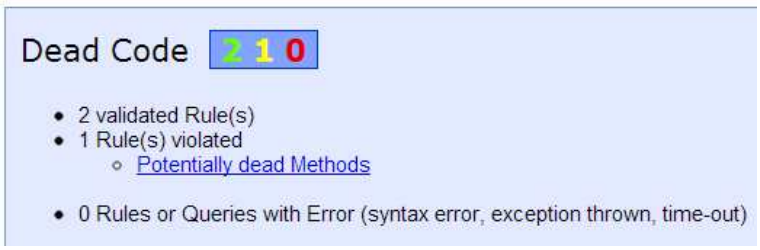
method	# Variables	Full Name
Interface()	41	Interface.Interface()
method	# Variables	Full Name

Showing 1 to 1 of 1 entries First Previous 1 Next Last

Statistics

Stat	# Variables
Sum:	41
Average:	41
Minimum:	41
Maximum:	41
Standard deviation:	0
Variance:	0

해당 Method 들은 사용하고 있는 변수를 축소하는 것을 권장하고 있다. 15 개의 이상의 변수를 사용하고 있을 시에, 해당 Method 를 수정할 때의 Side-Effect 가 큰 편이기 때문에 해당 위험도를 낮추기 위한 권장 사항으로 보인다.



Dead Code 2 1 0

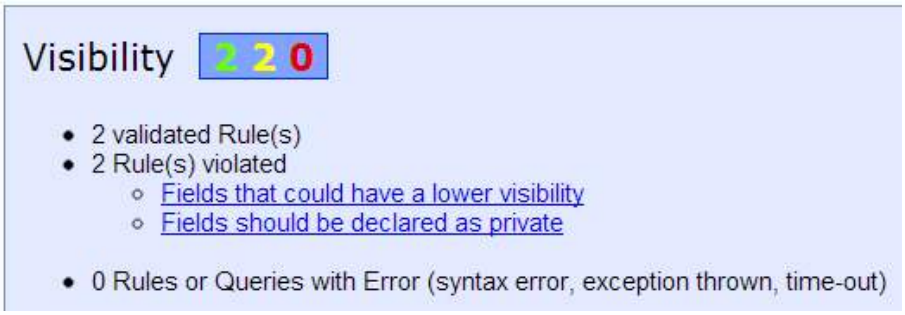
- 2 validated Rule(s)
- 1 Rule(s) violated
  - Potentially dead Methods
- 0 Rules or Queries with Error (syntax error, exception thrown, time-out)

3.2.7 Potentially dead Methods

Show 25 entries Search:

methods	MethodsCallingMe	depth	Full Name
getTransferData(DataFlavor)	0 method	0	AreaTool$CopyImagetoClipBoard$TransferableImage.getTransferData (DataFlavor)
getTransferDataFlavors()	1 method	1	AreaTool$CopyImagetoClipBoard$TransferableImage.getTransferDataFlavors ()
isDataFlavorSupported(DataFlavor)	0 method	0	AreaTool$CopyImagetoClipBoard$TransferableImage.isDataFlavorSupported (DataFlavor)

해당 Method 들은 사용되지 않은 Method 로 이는 해당 프로젝트의 낭비가 될 수 있으므로 확인하는 것을 권장한다.



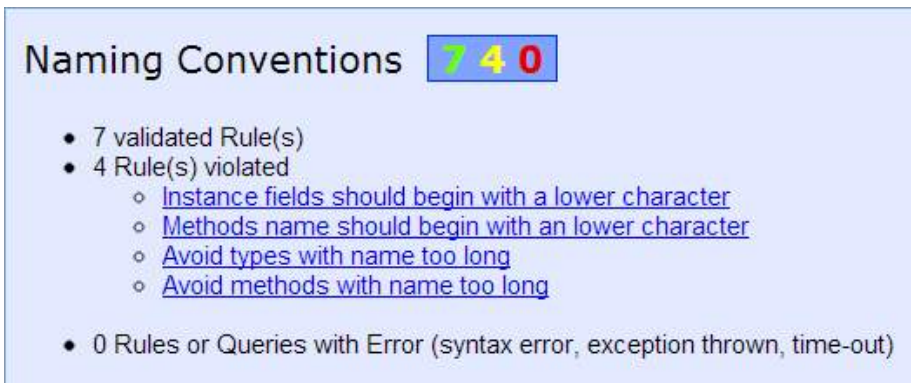
3.2.8 Fields That Could Have A Lower Visibility

fields	Size of instance	Full Name
i	4	AreaTool$CopyImagetoClipBoard$TransferableImage.i
CB	4	AreaTool.CB
color	4	Brush.color
img	4	Cut.img
cutnum	4	Cut.cutnum
width	4	Cut.width
height	4	Cut.height
cutstat	4	Cut.cutstat
brush	4	DrawingTool.brush
eraser	4	DrawingTool.eraser
linesize	4	DrawingTool.linesize
x	4	Interface$5.x
y	4	Interface$5.y
SelectedTool	4	Interface.SelectedTool
pj	4	Interface.pj
dt	4	Interface.dt
EnabledCutNum	4	Interface.EnabledCutNum
at	4	Interface.at
CanvasPanel	4	Interface.CanvasPanel
CanvasList	4	Interface.CanvasList
size	4	Interface.size
color	4	Interface.color
CutListPanel	4	Interface.CutListPanel

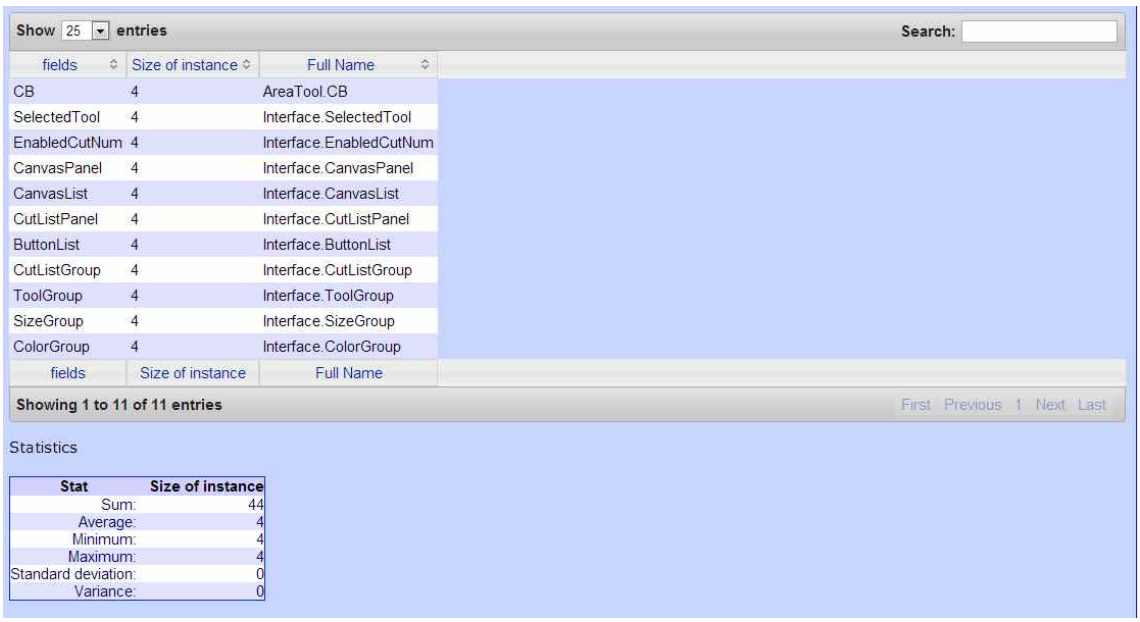
3.2.9 Fields Should Be Declared As Private

fields	Visibility	CouldBeDeclared	MethodsUsingMe	Full Name
i	Public	Private	2 methods	AreaTool$CopyImagetoClipBoard$TransferableImage.i
CB	Public	Private	1 method	AreaTool.CB
color	Public	Internal	5 methods	Brush.color
img	Public	Internal	12 methods	Cut.img
cutnum	Public	Internal	2 methods	Cut.cutnum
width	Public	Internal	8 methods	Cut.width
height	Public	Internal	9 methods	Cut.height
cutstat	Public	Internal	11 methods	Cut.cutstat
brush	Public	Internal	5 methods	DrawingTool.brush
eraser	Public	Private	1 method	DrawingTool.eraser
linesize	Public	Internal	8 methods	DrawingTool.linesize
x	Public	Private	1 method	Interface$5.x
y	Public	Private	1 method	Interface$5.y
SelectedTool	Public	Internal	17 methods	Interface.SelectedTool
pj	Public	Internal	23 methods	Interface.pj
dt	Public	Internal	8 methods	Interface.dt
EnabledCutNum	Public	Internal	16 methods	Interface.EnabledCutNum
at	Public	Internal	19 methods	Interface.at
CanvasPanel	Public	Internal	7 methods	Interface.CanvasPanel
CanvasList	Public	Internal	10 methods	Interface.CanvasList
size	Public	Internal	5 methods	Interface.size
color	Public	Private	1 method	Interface.color
CutListPanel	Public	Private	3 methods	Interface.CutListPanel
ButtonList	Public	Internal	6 methods	Interface.ButtonList
CutListGroup	Public	Private	2 methods	Interface.CutListGroup

위의 변수들을 Private 로 선언해주는 것을 권장하고 있다. (참고)



3.2.10 Instance Fields Should Begin With A Lower Character



Show 25 entries Search:

fields	Size of instance	Full Name
CB	4	AreaTool.CB
SelectedTool	4	Interface.SelectedTool
EnabledCutNum	4	Interface.EnabledCutNum
CanvasPanel	4	Interface.CanvasPanel
CanvasList	4	Interface.CanvasList
CutListPanel	4	Interface.CutListPanel
ButtonList	4	Interface.ButtonList
CutListGroup	4	Interface.CutListGroup
ToolGroup	4	Interface.ToolGroup
SizeGroup	4	Interface.SizeGroup
ColorGroup	4	Interface.ColorGroup
fields	Size of instance	Full Name

Showing 1 to 11 of 11 entries First Previous 1 Next Last

Statistics

Stat	Size of instance
Sum:	44
Average:	4
Minimum:	4
Maximum:	4
Standard deviation:	0
Variance:	0

3.2.11 Methods Name Should Begin With An Lower Character

Show 25 entries Search:

methods	Full Name
LoadCut(String)	Cut.LoadCut(String)
SaveCut(String)	Cut.SaveCut(String)
methods	Full Name

Showing 1 to 2 of 2 entries First Previous 1 Next Last

Statistics

Stat
Sum:
Average:
Minimum:
Maximum:
Standard deviation:
Variance:

해당 Instance Fields 와 Methods Name 에는 소문자를 쓰는 것을 권장하고 있다.

4. Ant, CruiseControl

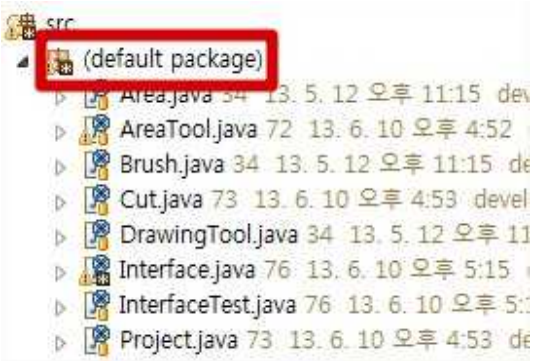
```
================================================================
<?xml version="1.0"?>

<project default="run" basedir=".">

        <echo message="pulling in property files"/>
        <property file="axis_bujava.properties"/>
        <property name="build.dir"                      location="build"/>
        <property name="build.main.dir"     location="${build.dir}/main"/>
        <property name="build.test.dir"             location="${build.dir}/test"/>
        <property name="dest.dir"                       location="dest"/>
        <property name="src.dir"                        location="src"/>
        <property name="lib.dir"
       location="C:₩Users₩Soulkey₩workspace₩lib"/>
        <path id="project.classpath">
                <pathelement location="${build.main.dir}"/>
                <fileset dir="${lib.dir}">
                        <include name="**/*.jar"/>
                </fileset>
        </path>

        <target name="clean">
                <delete dir="${build.main.dir}"/>
                <delete dir="${build.test.dir}"/>
                <delete dir="${dest.dir}"/>
        </target>
        <target name="prepare" depends="clean">
                <mkdir dir="${build.main.dir}"/>
                <mkdir dir="${build.test.dir}"/>
                <mkdir dir="${dest.dir}"/>
        </target>
        <target name="compile" depends="prepare">
                <javac srcdir="${src.dir}" destdir="${build.main.dir}" includeantruntime="false">
                        <exclude name="**/*Test.java"/>
                        <classpath refid="project.classpath"/>
                </javac>
```

```
	</target>
	<target name="compile-test" depends="compile">
		<javac srcdir="${src.dir}" destdir="${build.test.dir}" includeantruntime="false">
			<include name="**/*Test.java"/>
			<classpath refid="project.classpath"/>
		</javac>
	</target>

	<!-- test Target 를 주석처리한 이유는 해당 테스트케이스 중 LoadImage 는 사용자가 개입되어야해서 Ant 상으로 불가능함>
	<target name="test" depends="compile-test">
		<junit fork="true" haltonfailure="true" printsummary="true">
			<classpath refid="project.classpath"/>
			<formatter type="xml"/>
			<batchtest>
				<fileset dir="${src.dir}">
					<include name="**/*Test.java"/>
				</fileset>
			</batchtest>
		</junit>
	</target>
	<-->

	<!-- test javadoc 을 주석처리한 이유는 해당 프로젝트의 package 이름이 특정되어있지 않기때문에 에러가 발생>
	<target name="javadoc" depends="compile-test">
		<mkdir dir="${docs.dir}"/>
		<javadoc sourcepath="${src.dir}" destdir="${docs.dir}" packagenames="*">
			<classpath refid="project.classpath"/>
		</javadoc>
	</target>
	<-->

	<target name="run" depends="compile-test">
		<java classname="Interface" fork="yes" classpath="${build.main.dir}" classpathref="project.classpath"/>
	</target>

</project>
```

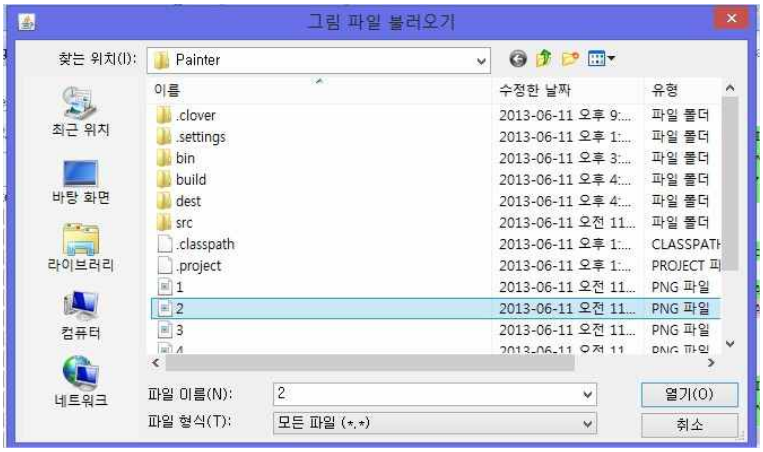
==========

해당 프로젝트를 진행하는 동안, SMA Team2 는 Eclipse 빌드 시스템에 의존하여 CruiseControl 이 제공하는 Test 와 Error, Failure Report 에 대한 정보를 공유하기는 어려웠다. 특히 해당 프로젝트는 Ant 의 빌드 방식을 적용하기 어려웠다.

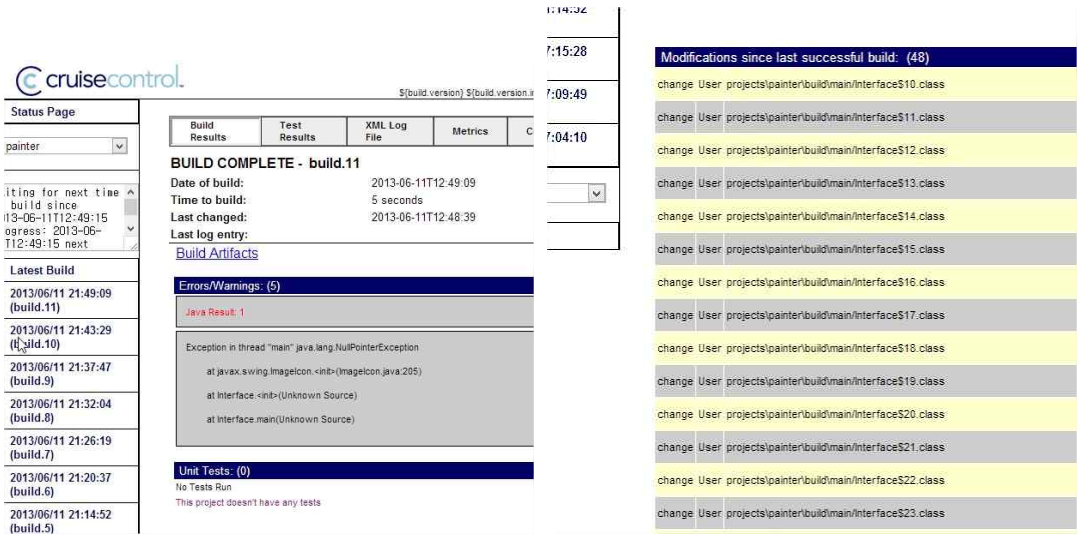
첫 번째로 해당 프로젝트를 위한 Test Case 를 작성한 Class 가 한 개에 불과해서 해당 클래스 파일을 일반 Source Code 파일에 함께 넣어뒀는데, 이 때문에 Ant Build 파일 작성에 있어서 Test 를 진행하는 쪽에서 이를 따로 분류해서 컴파일하고 그 결과를 따로 저장하는데 어느 정도 불편함이 있었다. 하지만 이런 문제는 처음 Ant Build 파일을 생성할 때만 문제가 되는 것이어서, Mantis 에서도 이에 대한 문제점을 언급하지 않았다.



두 번째 문제는 해당 프로젝트의 패키지가 Specify 가 되지 않은 (Default Package)라는 점이었다. Test 를 진행하면서 이러한 점은 큰 문제가 되지 않을 것이라고 생각했지만, Ant Build File 를 작성하는 도중, javadoc API 를 작성하는 과정에 있어서 패키지명이 Specify 되지 않아서 Failure 가 발생하는 문제점이 생겼다. 그러나 이러한 문제점 역시 CruiseControl 의 Config 를 통하여 자동적으로 jar 배포파일을 생성하도록 함으로써 배포 파일 빌드에 대한 문제점을 해결하였다.



세 번째 문제는 어느 정도 Critical 한 문제였는데, SMA Team2 에서 작성된 JUnit Test Code 중 LoadImage 에 대한 Test Case 는 사용자가 직접 파일을 로드해야만 Test 가 진행되었는데, 이는 자동화 빌드를 추구하는 Ant 와 CruiseControl 에 상충되는 개념이어서 문제가 발생하였다. 실제로 CruiseControl 에 올라온 Report 에서는 Test Case 가 전혀 수행되지 않은 것으로 나타났다. Eclipse 라는 IDE 에서 수행한 JUnit Test 를 통해 테스트 결과를 확인할 수 있었지만, 자동화된 빌드와 Unit Test 를 수행할 수 없던 것이 아쉬웠다.

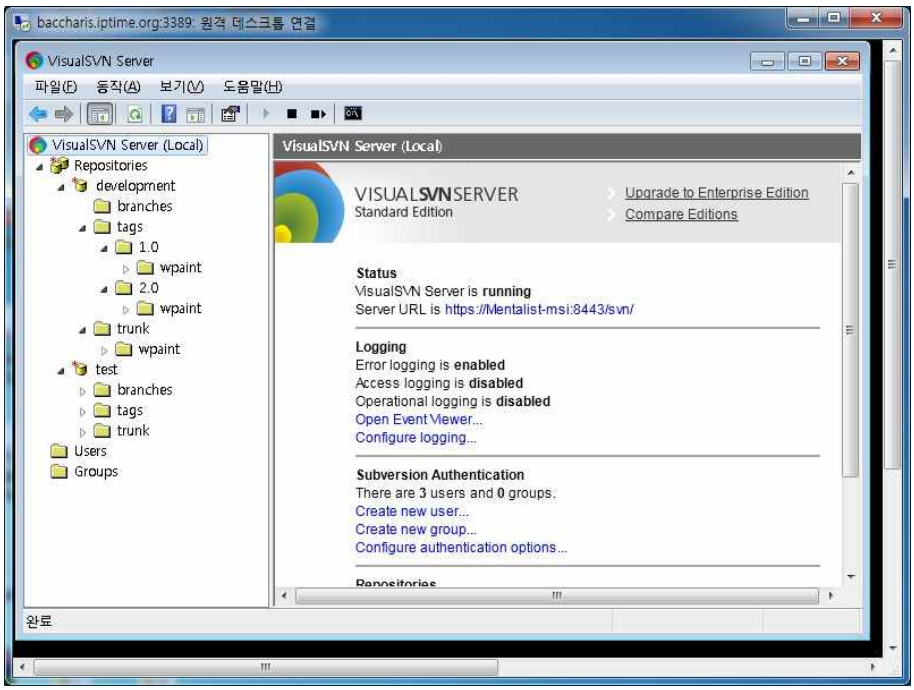


Ant, CruiseControl 은 개발자와 관리자, 사용자가 Eclipse 와 같은 IDE 를 굳이 켜서 테스트를 진행하지 않아도 자동으로 Unit Test 결과와 Build 결과를 확인할 수 있는 간편한 Tool 이었지만, Ant Build File 과 CruiseControl Configuration 파일 설정이 너무 어려워서 이에 대한 SMA Team2 과의 의견 교환이 어려웠고, 배포 파일과 Unit Test 가 쉽게 이루어지지 않았다는 점이 아쉬웠다.

5. SVN, Mantis

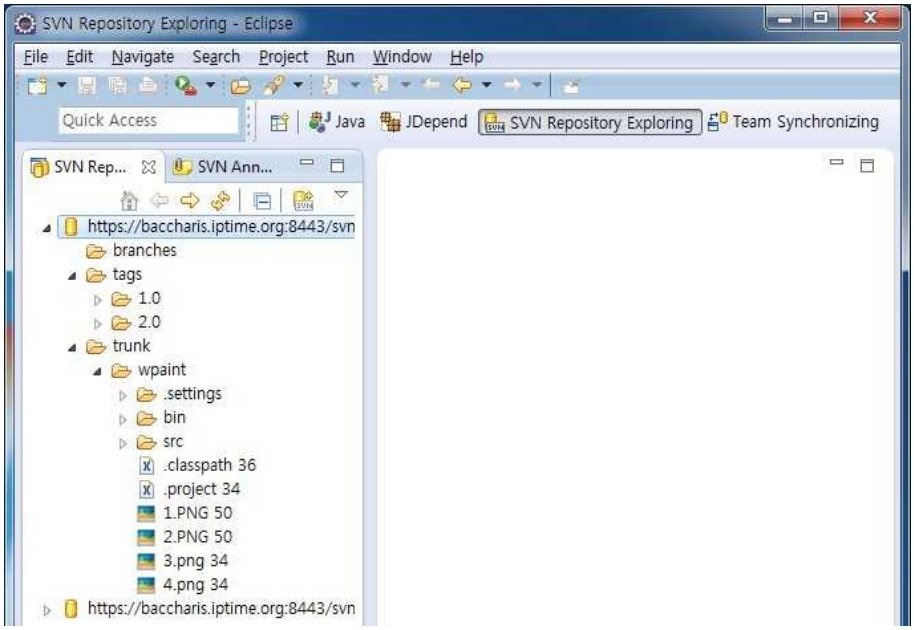
5.1 SVN

5.1.1 SVN 사용



우리는 Server 에서 Visual SVN 을 사용하고, Client Eclipse 에서 Subclipse 또는 Subversive – SVN Team Provider 등을 사용하였다. Subversion 은 3 학년 Team2 에서 개발하기 위해서 우리가 Test 하기 위해서 Code 를 공유하기 위해서 사용하였다. 첫 번째 발표 때 System Test 를 마치고 3 학년 Team2 에 Release 1.0 생성을 요구하였고, 두 번째 발표 때 System Test 와 Static Analysis 를 마치면서 Release 2.0 생성을 요구하였다. 그리고 한 번 더 Test 과정을 거치면서 수정이 완료되면 Release 3.0 을 생성해달라고 요구한 상태이다. 우리 팀에서는 Commit 을 할 필요가 없었고 Subversion 에서 Checkout 을 통해서 Project 전체를 가져오거나 Update 를 통해서 기존의 Code 를 3 학년 Team2 에서 수정한 상태로 바꾸었다.

SVN 을 사용하는 과정 속에서 큰 문제는 없었으나 처음에 System Test 를 수행할 때 3 학년 Team2 에서 최신의 수행 가능한 개발된 Code 를 branches 에 두었기 때문에 trunk 에서 많은 기능이 구현되지 않은 상태로 System Test 를 수행하는 어려움이 있었다. 그 후에는 Mantis 를 통해서 최신의 수행 가능한 개발된 Code 를 trunk 에 둘 것을 요구했고 그 이후로는 정상적으로 최신의 수행 가능한 개발된 Code 에 대해서만 Test 가 수행되었다.

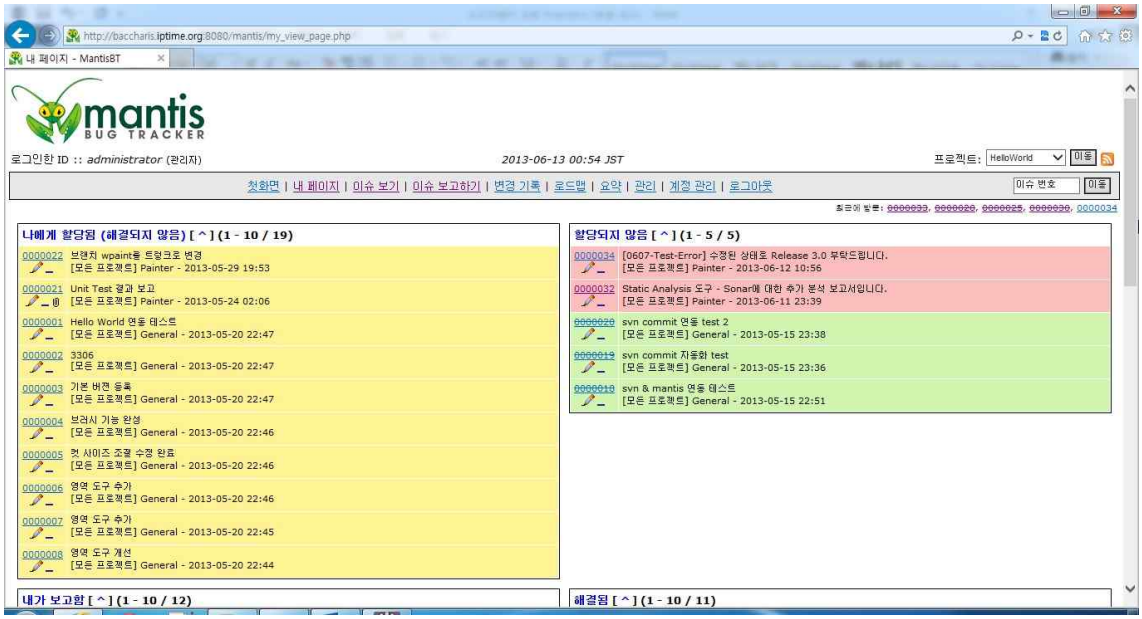


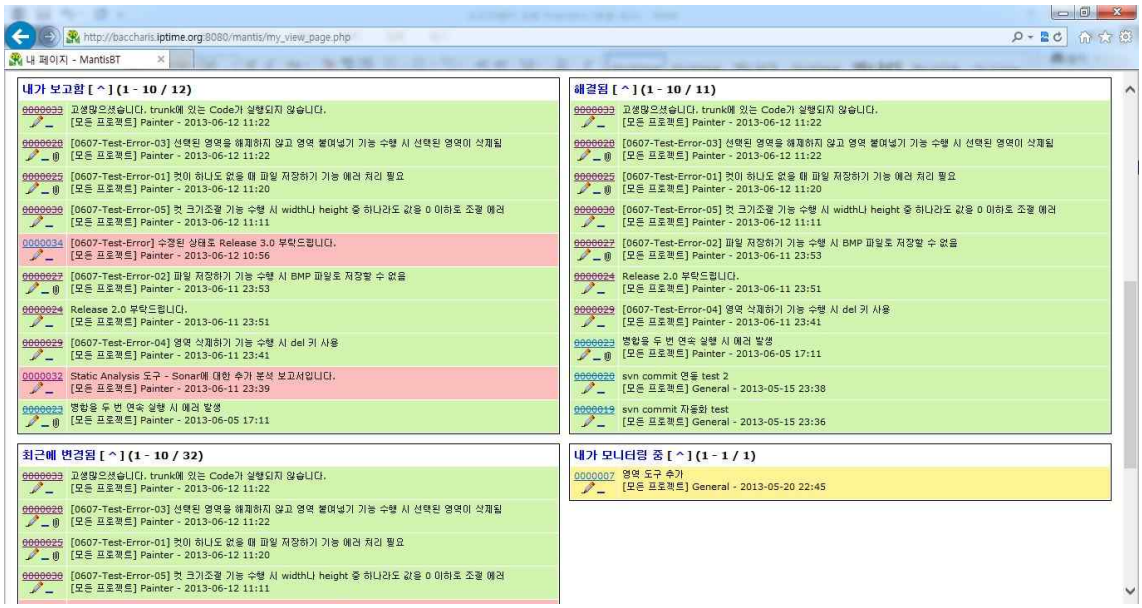
5.1.2 SVN 사용 하면서...

Subversion 을 처음 써보는 것은 아니었지만 branches, tag, trunk 등으로 각 기능을 하는 폴더별로 세분화 해서 사용했던 것은 처음이었다. tag 를 통해서 일정 기간마다 완성된 버전을 유지시킬 수 있었고, 개발은 branches 에서 따로 이루어지면서 trunk 를 통한 프로그램 Test 에 아무런 문제가 없었다. 개발자들 사이에서 같은 상태의 Code 를 쉽게 공유할 수 있고, Test 를 수행하는 우리 팀 같은 경우에 개발 과정과 독립적으로 Test 를 수행할 수 있다는 점에서 좋았다. 하지만 우리 팀은 Test 만 수행하는 경우라서 개발자들에 비해서 Subversion 의 장점을 많이 느끼지는 못했던 것 같다.

5.2 Mantis

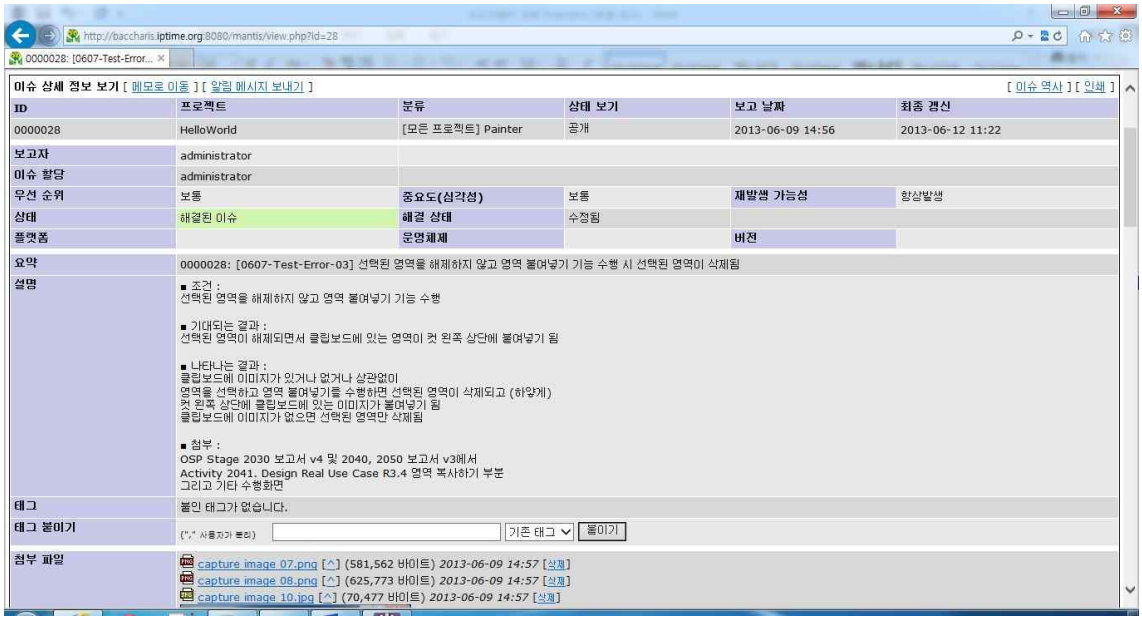
5.2.1 Mantis 사용

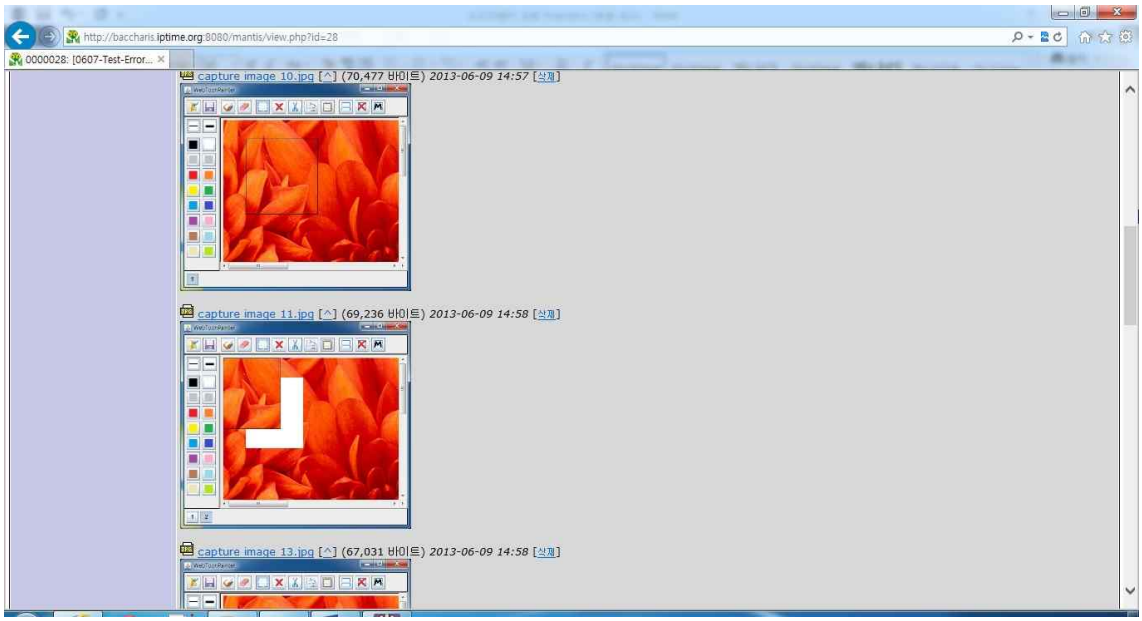




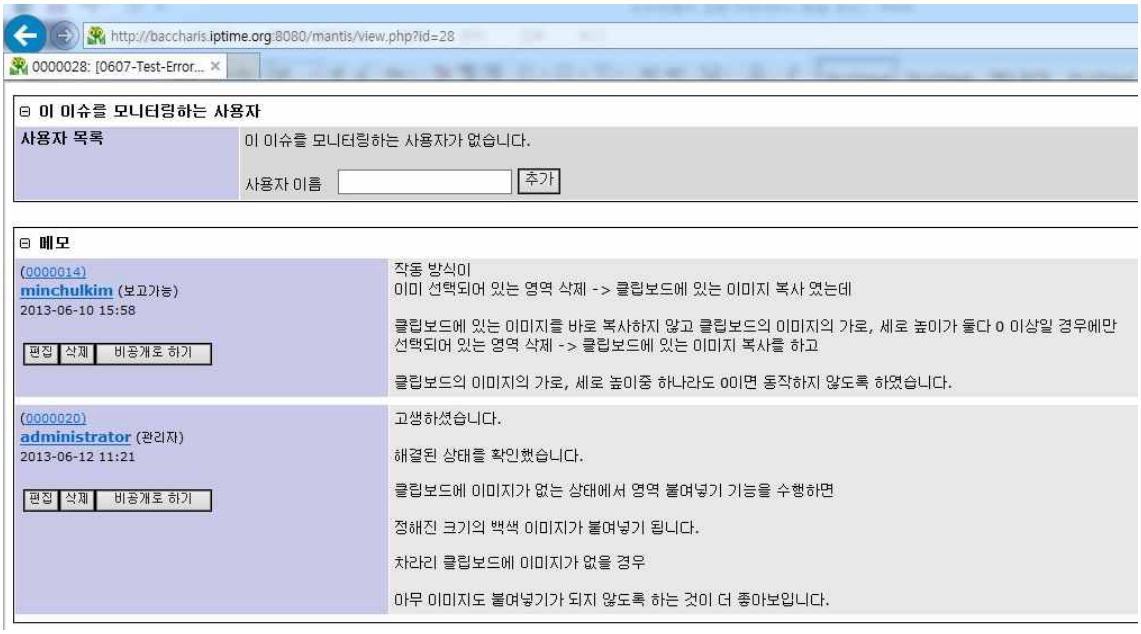
3 학년 Team2 와 커뮤니케이션을 위에서 Mantis 를 많이 사용하였다. 3 학년 Team2 에서는 개발을 하면서 기능을 구현할 때마다 Mantis 에 이슈를 남겨주었고, 우리 팀은 Test 중에 Error 를 발견할 때마다 정리해서 이슈를 남겼다. 전체적인 보고서는 과목 홈페이지에 올라가기 때문에 이슈에는 개선이 필요한 기능들의 명칭 정도만 나열하는 수준으로 작성하였다. 이렇게 이슈를 작성했을 때 커뮤니케이션에 문제가 없었고, 3 학년 Team2 에서도 기능이 개선이 되면 이슈에 메모를 남겨서 기능이 개선되었음을 알려주었다. 3 학년 Team2 에서 기능이 개선되었다고 알려준

이슈에 대해서는 다시 Test 를 수행하고 Error 가 해결되었을 때 '해결된 이슈' 상태로 바꾸어서 3 학년 Team2 가 Mantis 페이지를 봤을 때 알 수 있도록 하였다.

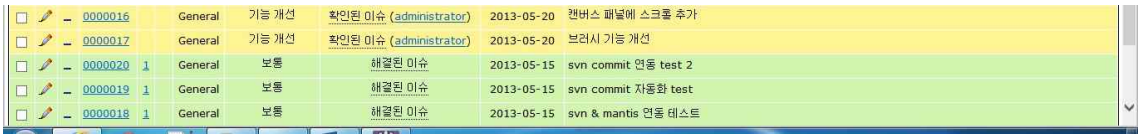




3 학년 Team2 에 Error 가 발생하는 기능에 대한 개선을 요구할 때 위와 같은 형식으로 이슈를 작성하였다. Error 가 발생하는 기능 당 이슈를 하나씩 작성해서 이외에도 이슈가 더 있지만 이슈 작성 형식을 소개하는 차원에서 하나의 이슈 화면만 붙여 넣었다. 이슈에는 Error 가 발생하는 조건과 기대되는 결과, 나타나는 결과, 그리고 Error 화면의 스크린 샷을 첨부하였다. 이렇게 이슈를 작성해서 3 학년 Team2 와 커뮤니케이션의 문제가 없었고, 과목 홈페이지에 보고서도 참고할 수 있었다.



3 학년 Team2 에서 기능 개선이 완료되면 Error 해결 여부와 어떻게 해결했는지 등을 이슈에 메모로 남겨주었고, 우리 팀도 기능 개선이 완료된 기능에 대해 다시 한 번 테스트를 수행한 후에 그 결과에 대해 Comment 를 작성하였다.



Mantis 와 Subversion 연동이 이루어졌고 3 학년 Team2 와도 커뮤니케이션을 했지만 Mantis 와 Subversion 의 연동이란 Mantis 이미 작성된 이슈에 대해 Subversion 에서 Commit 시 이미 알고 있는 이슈번호를 지정해서 Commit Comment 를 이슈의 상태 변경과 함께 Comment 하는 것이기 때문에 굳이 연동을 하지 않고 Mantis 에 새 이슈를 작성해서 개발자와 테스터들이 서로 확인하는 것과 다르지 않았다. 그래서 연동은 해놓은 상태로 Subversion Commit Comment 로 이슈를 변경하지 않고 직접 이슈를 새로 작성하는 방향으로 진행하였다.

5.2.2 Mantis 사용 하면서...

처음에는 굳이 Mantis 를 왜 사용하나 싶었지만 3 학년 Team2 와 하나, 하나 전화나 문자로 커뮤니케이션 하는 것보다 편하다는 것을 느끼고 끝 무렵에 가서 많이 사용하였다. 작성된 이슈에 대해 3 학년 Team2 에서 메모를 작성할 경우 메일로 알려주어서 굳이 Mantis 에 들어가서 확인해보지 않더라도 개선된 내용을 알 수 있어서 편리함을 느꼈다. 전화나 문자를 사용하는 경우에는 서로 커뮤니케이션을 나눴던 내용이 분산되거나 사라지는 경우가 많은데 Mantis 를 사용하게 되면 전부 Mantis 홈페이지에 남아있기 때문에 지금까지 어떤 일을 해왔는지, 앞으로 어떤 일들을 해야 하는지 쉽게 파악할 수 있었다. 물론 개발 과정 속에서 Test 과정 속에서

Mantis 에 이슈를 작성해야 하는 일이 늘었지만 개발을 활발하게 진행하기 위해서 의미 있는 일이라고 생각된다.

기능 개선을 요구하는 이슈를 작성하고 3 학년 Team2 에서 기능 개선을 완료하고 메모를 달았을 때 받은 메일이다. 굳이 Mantis 에 들어가보지 않고도 3 학년 Team2 의 진행 상황을 알 수 있어서 좋았다.