Team3.
Sangyoon Kim.
Green Kim.

Final Presentation

1.CTIP

2.System Testing

3.Static Analysis

CONTENTS

1. CTIP

1. Activity, Problem and Result
2. What we've learned and Impression

2. System Testing

1. Activity, Problem and Result
2. What we've learned and Impression

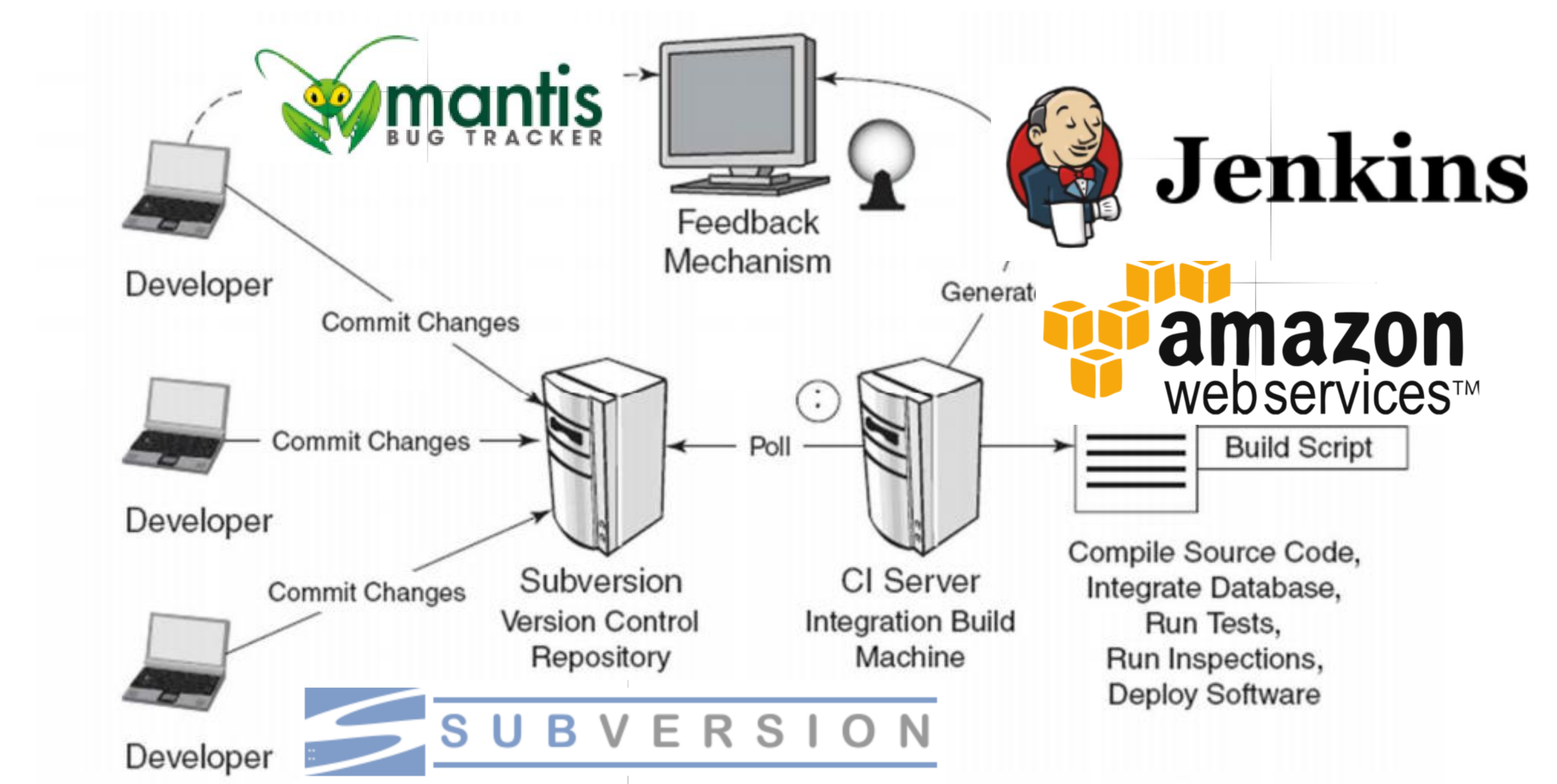
3. Static Analysis

1. Activity, Problem and Result
2. What we've learned and Impression

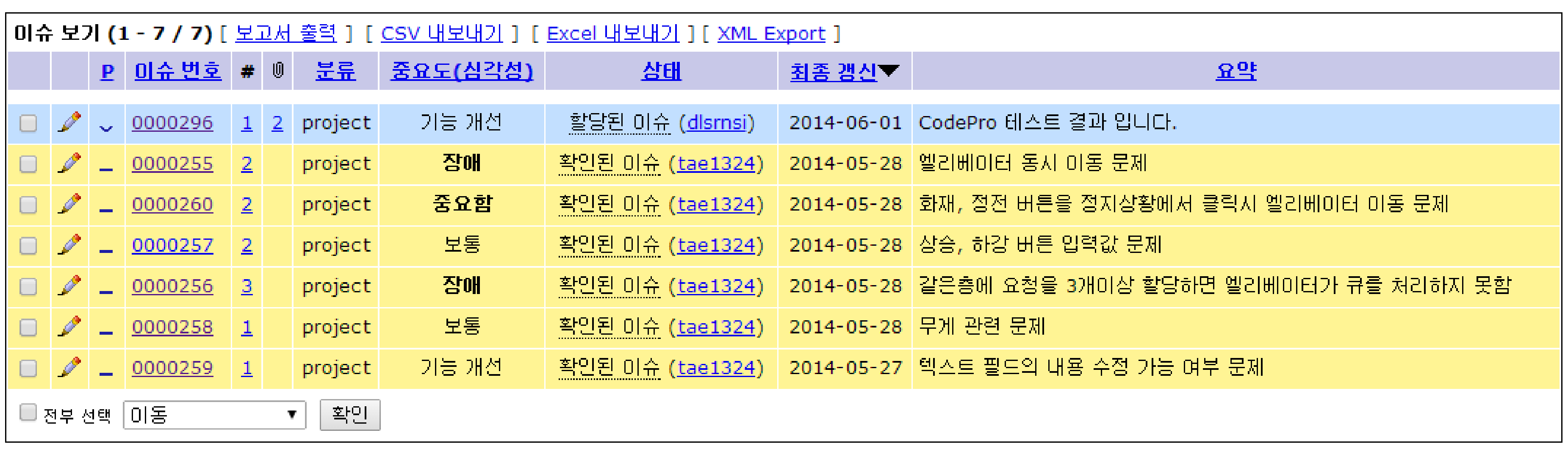
1. CTIP

1. Activity, Problem and Result
2. What we’ve learned and Impression

1. Activity, Problem and Result



1. Activity, Problem and Result
Mantis



이슈 보기 (1 - 7 / 7) [보고서 출력] [CSV 내보내기] [Excel 내보내기] [XML Export]

		P	이슈 번호	#	📎	분류	중요도(심각성)	상태	최종 갱신▼	요약
□	✎	⌄	0000296	1	2	project	기능 개선	할당된 이슈 (dlsrnsi)	2014-06-01	CodePro 테스트 결과 입니다.
□	✎	_	0000255	2		project	**장애**	확인된 이슈 (tae1324)	2014-05-28	엘리베이터 동시 이동 문제
□	✎	_	0000260	2		project	**중요함**	확인된 이슈 (tae1324)	2014-05-28	화재, 정전 버튼을 정지상황에서 클릭시 엘리베이터 이동 문제
□	✎	_	0000257	2		project	보통	확인된 이슈 (tae1324)	2014-05-28	상승, 하강 버튼 입력값 문제
□	✎	_	0000256	3		project	**장애**	확인된 이슈 (tae1324)	2014-05-28	같은층에 요청을 3개이상 할당하면 엘리베이터가 큐를 처리하지 못함
□	✎	_	0000258	1		project	보통	확인된 이슈 (tae1324)	2014-05-28	무게 관련 문제
□	✎	_	0000259	1		project	기능 개선	확인된 이슈 (tae1324)	2014-05-27	텍스트 필드의 내용 수정 가능 여부 문제

□ 전부 선택 [이동 ▾] [확인]

2. What we've learned and Impression

1) Subversion
    1) Important of version management

2) Jenkins
    1) To check team working steady

2. What we’ve learned and Impression

3) Mantis
   1) Importance of Bug tracking
   2) Using APM
   3) Popularlity

4) Interlocking

Jenkins – Mantis : Error!

2. What we’ve learned and Impression

5) Pros and Cons of CTIP

Cons : CTIP is NOT PERFECT

1) Eclipse dependent Plugins
2) Lack of server
3) Developers mistake

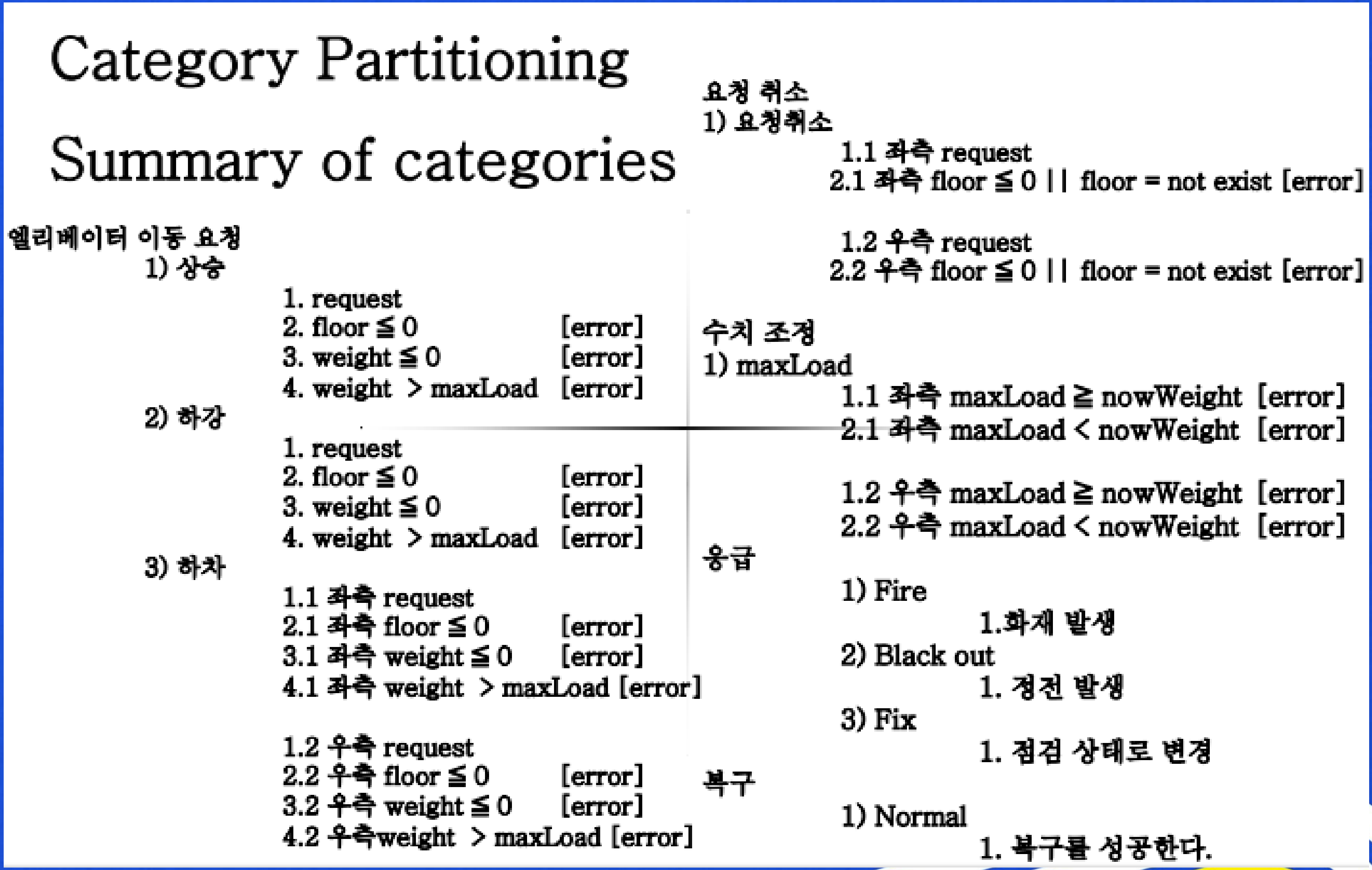
Pros : importance of Integration

1) Large scale Process
2) Consistency of source code

2. System Testing

1. Activity, Problem and Result
2. What we’ve learned and Impression

1. Activity, Problem and Result



Category Partitioning

Summary of categories

엘리베이터 이동 요청

1) 상승

1. request
2. floor ≦ 0 [error]
3. weight ≦ 0 [error]
4. weight > maxLoad [error]

2) 하강

1. request
2. floor ≦ 0 [error]
3. weight ≦ 0 [error]
4. weight > maxLoad [error]

3) 하차

1.1 좌측 request
2.1 좌측 floor ≦ 0 [error]
3.1 좌측 weight ≦ 0 [error]
4.1 좌측 weight > maxLoad [error]

1.2 우측 request
2.2 우측 floor ≦ 0 [error]
3.2 우측 weight ≦ 0 [error]
4.2 우측weight > maxLoad [error]

요청 취소

1) 요청취소

1.1 좌측 request
2.1 좌측 floor ≦ 0 || floor = not exist [error]

1.2 우측 request
2.2 우측 floor ≦ 0 || floor = not exist [error]

수치 조정

1) maxLoad

1.1 좌측 maxLoad ≧ nowWeight [error]
2.1 좌측 maxLoad < nowWeight [error]

1.2 우측 maxLoad ≧ nowWeight [error]
2.2 우측 maxLoad < nowWeight [error]

응급

1) Fire

1.화재 발생

2) Black out

1. 정전 발생

3) Fix

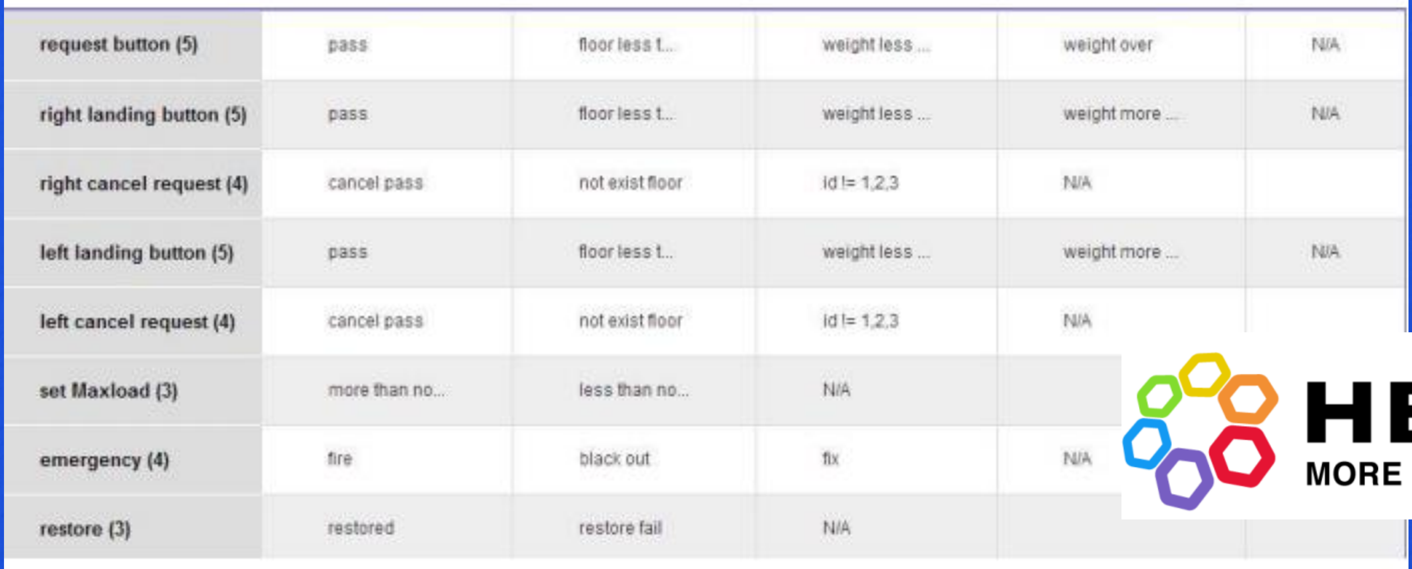
1. 점검 상태로 변경

복구

1) Normal

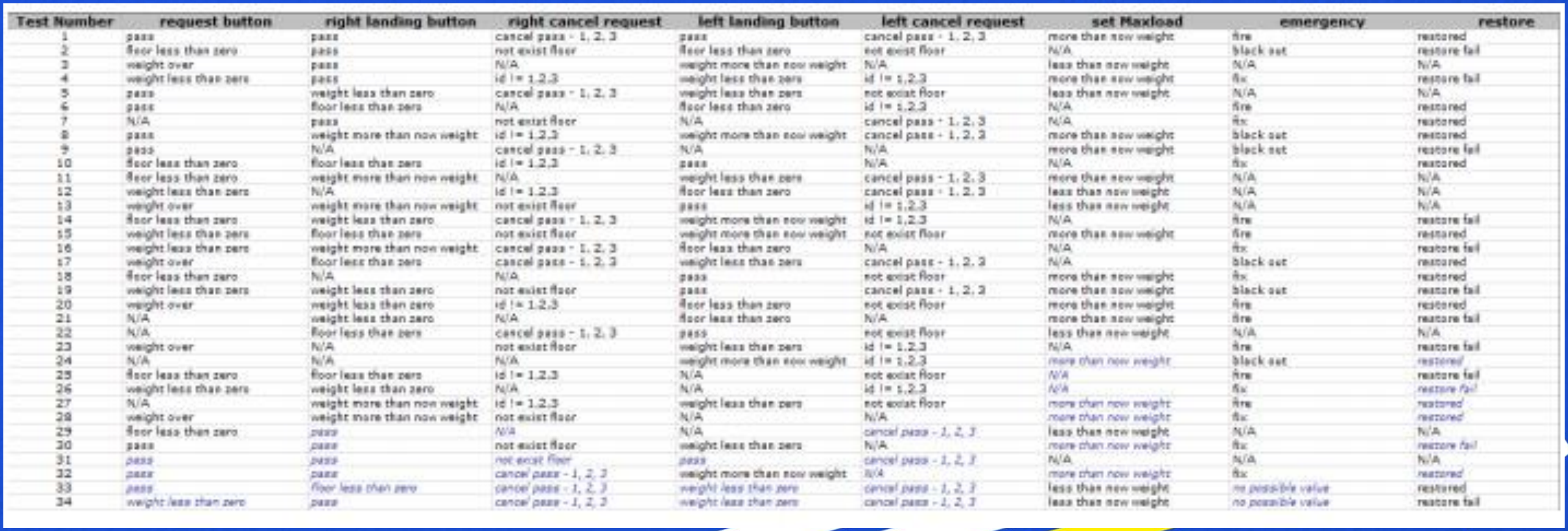
1. 복구를 성공한다.

1. Activity, Problem and Result



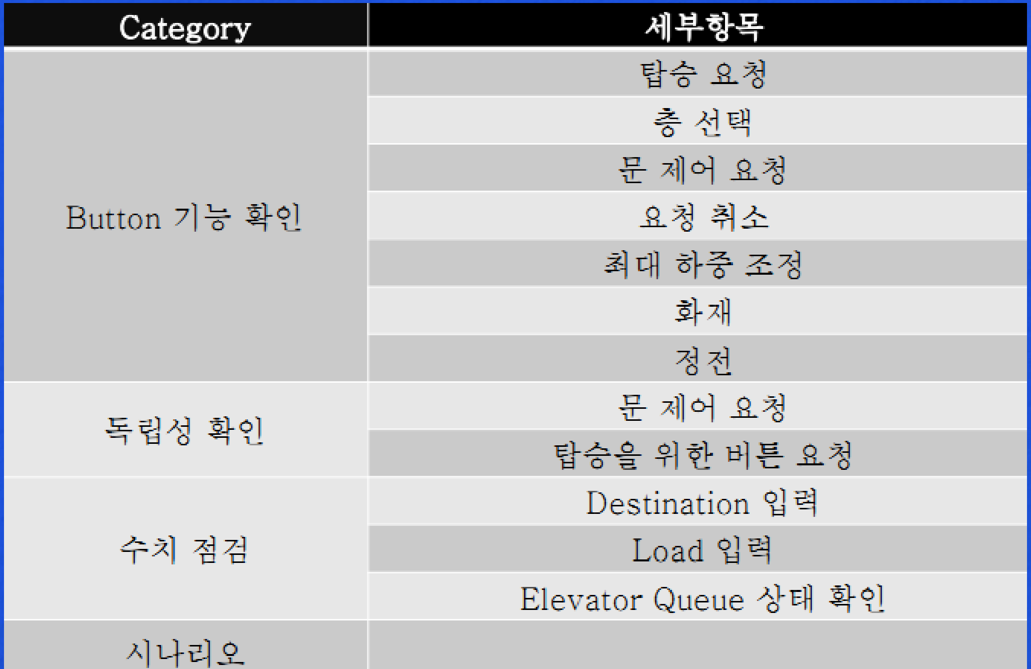
request button (5)	pass	floor less t...	weight less ...	weight over	N/A
right landing button (5)	pass	floor less t...	weight less ...	weight more ...	N/A
right cancel request (4)	cancel pass	not exist floor	id != 1,2,3	N/A	
left landing button (5)	pass	floor less t...	weight less ...	weight more ...	N/A
left cancel request (4)	cancel pass	not exist floor	id != 1,2,3	N/A	
set Maxload (3)	more than no...	less than no...	N/A		
emergency (4)	fire	black out	fix	N/A	
restore (3)	restored	restore fail	N/A		

HEXAWISE
MORE COVERAGE. FEWER TESTS.

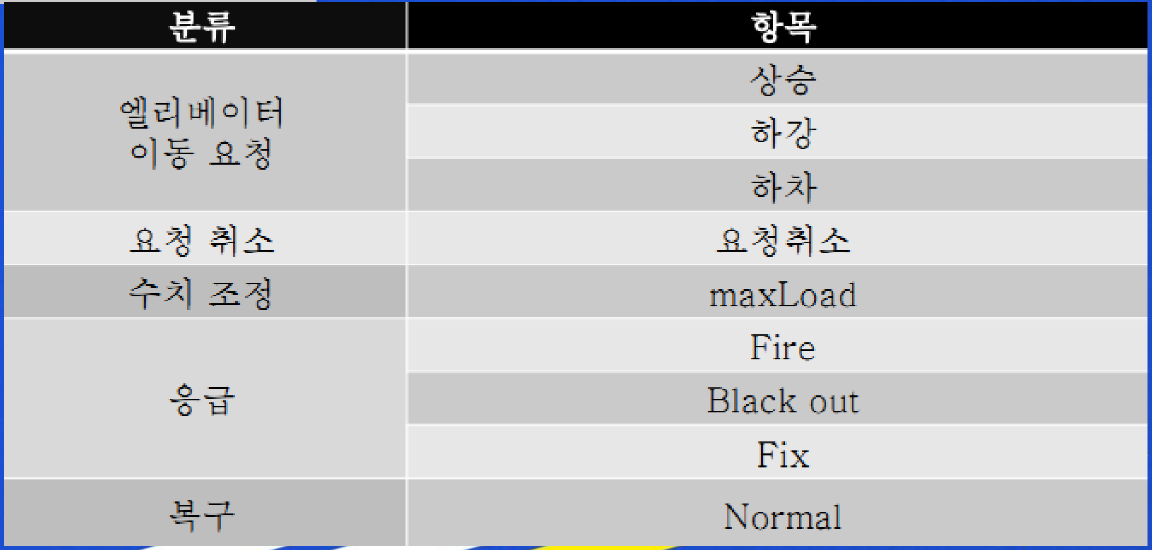


Test Number	request button	right landing button	right cancel request	left landing button	left cancel request	set Maxload	emergency	restore
1	pass	pass	cancel pass - 1, 2, 3	pass	cancel pass - 1, 2, 3	more than now weight	fire	restored
2	floor less than zero	pass	not exist floor	floor less than zero	not exist floor	N/A	black out	restore fail
3	weight over	pass	N/A	weight more than now weight	N/A	less than now weight	N/A	N/A
4	weight less than zero	pass	id != 1,2,3	weight less than zero	id != 1,2,3	more than now weight	fix	restore fail
5	pass	weight less than zero	cancel pass - 1, 2, 3	weight less than zero	not exist floor	less than now weight	N/A	N/A
6	pass	floor less than zero	N/A	floor less than zero	id != 1,2,3	N/A	fire	restored
7	N/A	pass	not exist floor	N/A	cancel pass - 1, 2, 3	N/A	fix	restored
8	pass	weight more than now weight	id != 1,2,3	weight more than now weight	cancel pass - 1, 2, 3	more than now weight	black out	restored
9	pass	N/A	cancel pass - 1, 2, 3	N/A	N/A	more than now weight	black out	restore fail
10	floor less than zero	floor less than zero	id != 1,2,3	pass	N/A	N/A	fix	restored
11	floor less than zero	weight more than now weight	N/A	weight less than zero	cancel pass - 1, 2, 3	more than now weight	N/A	N/A
12	weight less than zero	N/A	id != 1,2,3	floor less than zero	cancel pass - 1, 2, 3	less than now weight	N/A	N/A
13	weight over	weight more than now weight	not exist floor	pass	id != 1,2,3	less than now weight	N/A	N/A
14	floor less than zero	weight less than zero	cancel pass - 1, 2, 3	weight more than now weight	id != 1,2,3	N/A	fire	restore fail
15	weight less than zero	floor less than zero	not exist floor	weight more than now weight	not exist floor	more than now weight	fire	restored
16	weight less than zero	weight more than now weight	cancel pass - 1, 2, 3	floor less than zero	N/A	N/A	fix	restore fail
17	weight over	floor less than zero	cancel pass - 1, 2, 3	weight less than zero	cancel pass - 1, 2, 3	N/A	black out	restored
18	floor less than zero	N/A	N/A	pass	not exist floor	more than now weight	fix	restored
19	weight less than zero	weight less than zero	not exist floor	pass	cancel pass - 1, 2, 3	more than now weight	black out	restore fail
20	weight over	weight less than zero	id != 1,2,3	floor less than zero	not exist floor	more than now weight	fire	restored
21	N/A	weight less than zero	N/A	floor less than zero	N/A	more than now weight	fire	restore fail
22	N/A	floor less than zero	cancel pass - 1, 2, 3	pass	not exist floor	less than now weight	N/A	N/A
23	weight over	N/A	not exist floor	weight less than zero	id != 1,2,3	N/A	fire	restore fail
24	N/A	N/A	N/A	weight more than now weight	id != 1,2,3	*more than now weight*	black out	*restored*
25	floor less than zero	floor less than zero	id != 1,2,3	N/A	not exist floor	*N/A*	fire	restore fail
26	weight less than zero	weight less than zero	N/A	N/A	id != 1,2,3	*N/A*	fix	*restore fail*
27	N/A	weight more than now weight	id != 1,2,3	weight less than zero	not exist floor	*more than now weight*	fire	*restored*
28	weight over	weight more than now weight	not exist floor	N/A	N/A	*more than now weight*	fix	*restored*
29	floor less than zero	*pass*	*N/A*	N/A	*cancel pass - 1, 2, 3*	less than now weight	N/A	N/A
30	pass	*pass*	not exist floor	weight less than zero	N/A	*more than now weight*	fix	*restore fail*
31	*pass*	*pass*	*not exist floor*	*pass*	*cancel pass - 1, 2, 3*	N/A	N/A	N/A
32	*pass*	*pass*	*cancel pass - 1, 2, 3*	weight more than now weight	*N/A*	*more than now weight*	fix	*restored*
33	*pass*	*floor less than zero*	*cancel pass - 1, 2, 3*	*weight less than zero*	*cancel pass - 1, 2, 3*	less than now weight	*no possible value*	restored
34	*weight less than zero*	*pass*	*cancel pass - 1, 2, 3*	*weight less than zero*	*cancel pass - 1, 2, 3*	less than now weight	*no possible value*	restore fail

Big Mistake! : Categorization : Only 6144?



Category	세부항목
Button 기능 확인	탑승 요청
	층 선택
	문 제어 요청
	요청 취소
	최대 하중 조정
	화재
	정전
독립성 확인	문 제어 요청
	탑승을 위한 버튼 요청
수치 점검	Destination 입력
	Load 입력
	Elevator Queue 상태 확인
시나리오	



분류	항목
엘리베이터 이동 요청	상승
	하강
	하차
요청 취소	요청취소
수치 조정	maxLoad
응급	Fire
	Black out
	Fix
복구	Normal

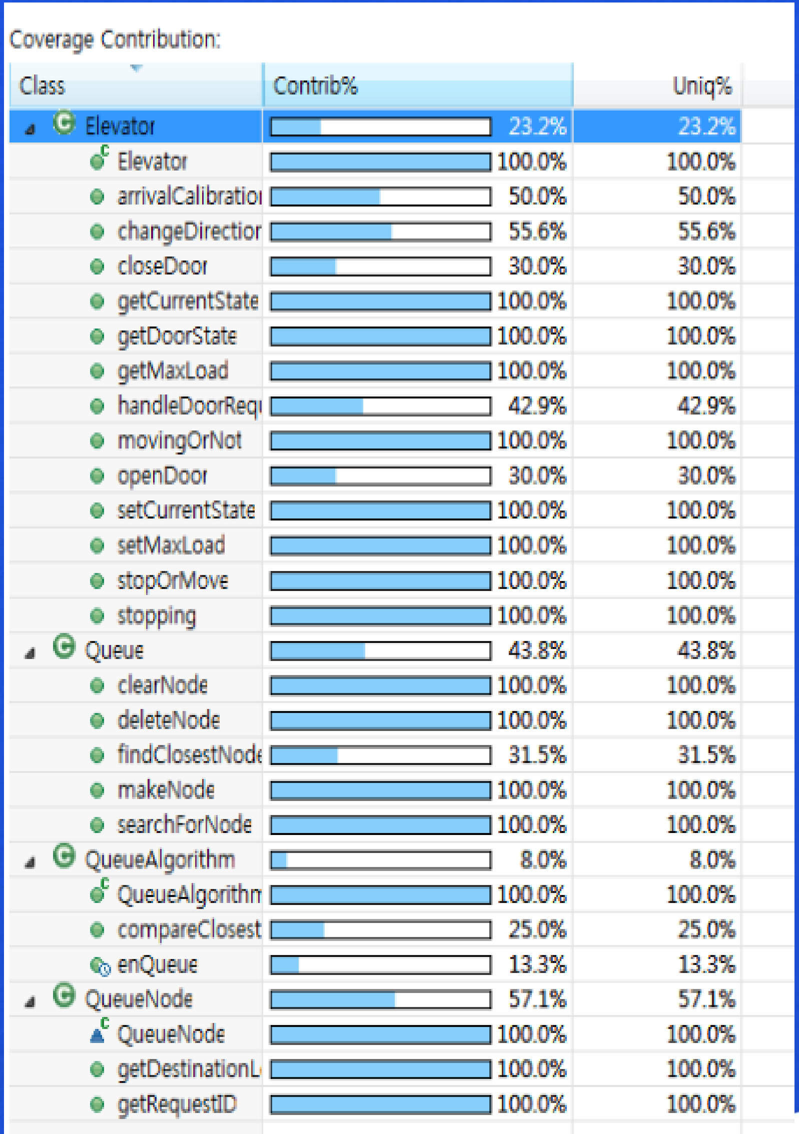
2. What we've learned and Impression

1) Importance of Requirement
2) Impossible to find all errors
3) Importance of Knowledge and Experience
4) Importance of Categorization

3. Static Analysis

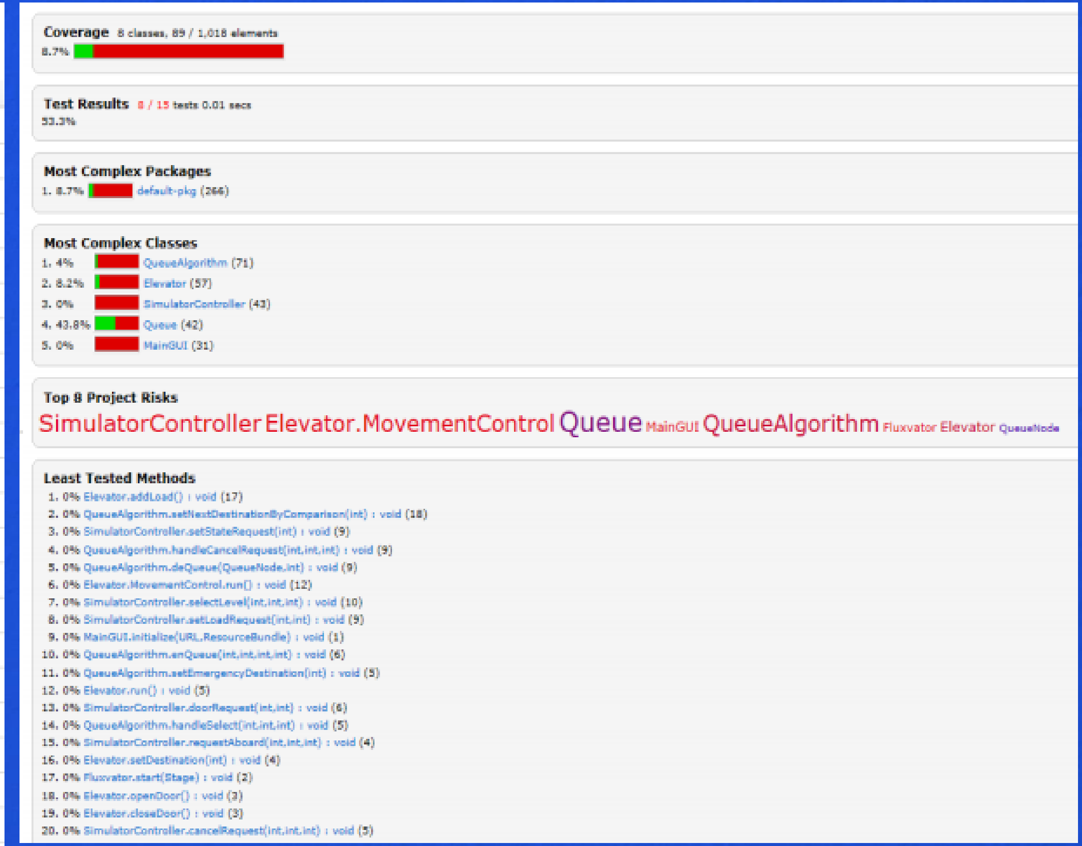
1. Activity, Problem and Result
2. What we've learned and Impression

1. Activity, Problem and Result
Clover : Coverage Contribution, Total Result



Coverage Contribution:

Class	Contrib%	Uniq%
Elevator	23.2%	23.2%
Elevator	100.0%	100.0%
arrivalCalibratio	50.0%	50.0%
changeDirectior	55.6%	55.6%
closeDoor	30.0%	30.0%
getCurrentState	100.0%	100.0%
getDoorState	100.0%	100.0%
getMaxLoad	100.0%	100.0%
handleDoorRequ	42.9%	42.9%
movingOrNot	100.0%	100.0%
openDoor	30.0%	30.0%
setCurrentState	100.0%	100.0%
setMaxLoad	100.0%	100.0%
stopOrMove	100.0%	100.0%
stopping	100.0%	100.0%
Queue	43.8%	43.8%
clearNode	100.0%	100.0%
deleteNode	100.0%	100.0%
findClosestNode	31.5%	31.5%
makeNode	100.0%	100.0%
searchForNode	100.0%	100.0%
QueueAlgorithm	8.0%	8.0%
QueueAlgorithm	100.0%	100.0%
compareClosest	25.0%	25.0%
enQueue	13.3%	13.3%
QueueNode	57.1%	57.1%
QueueNode	100.0%	100.0%
getDestinationL	100.0%	100.0%
getRequestID	100.0%	100.0%



Coverage 8 classes, 89 / 1,018 elements

8.7%

Test Results 8 / 15 tests 0.01 secs

53.3%

Most Complex Packages

1. 8.7% default-pkg (266)

Most Complex Classes

1. 4% QueueAlgorithm (71)
2. 8.2% Elevator (57)
3. 0% SimulatorController (43)
4. 43.8% Queue (42)
5. 0% MainGUI (31)

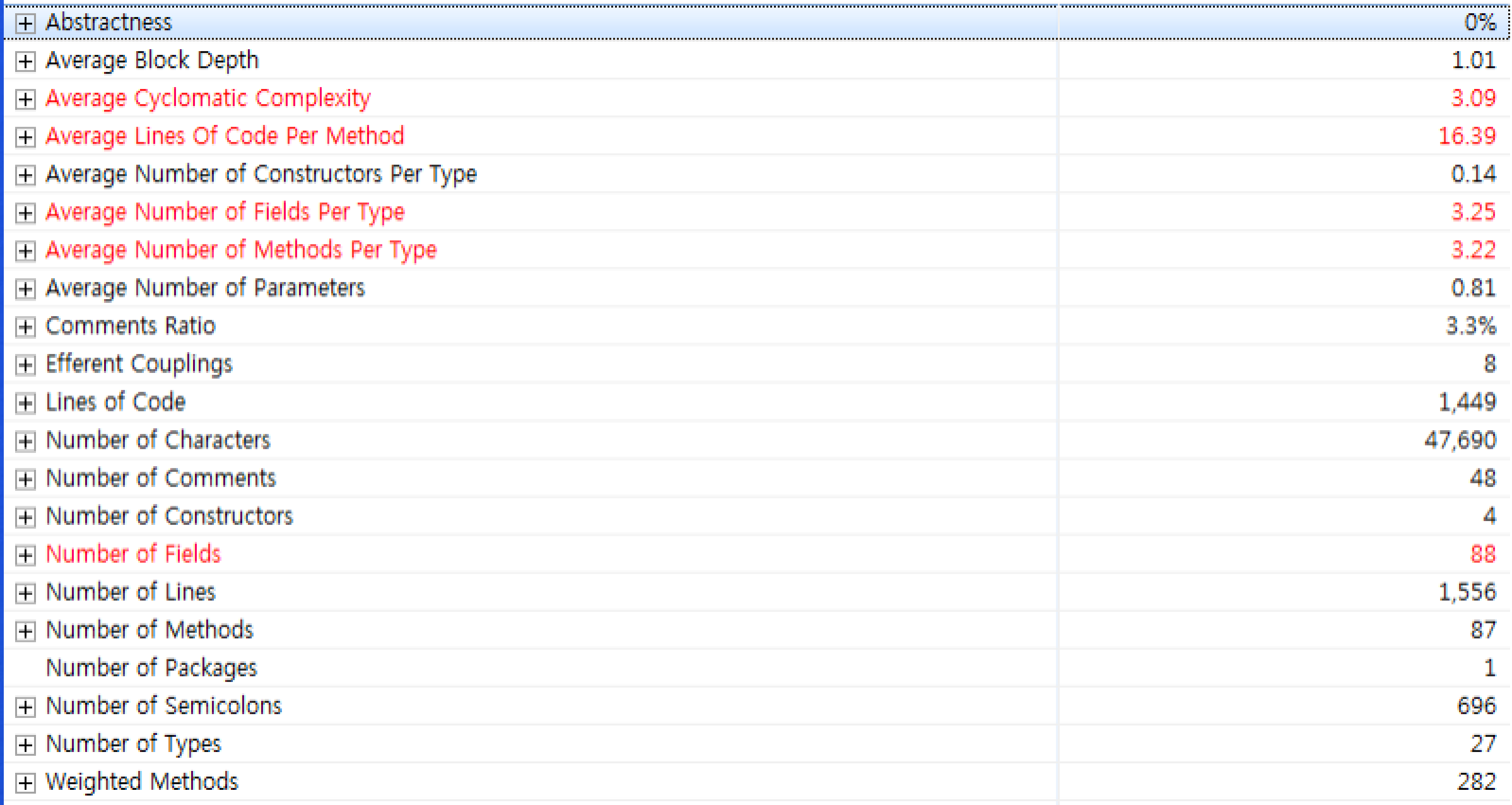
Top 8 Project Risks

SimulatorController Elevator.MovementControl Queue MainGUI QueueAlgorithm Fluxvator Elevator QueueNode

Least Tested Methods

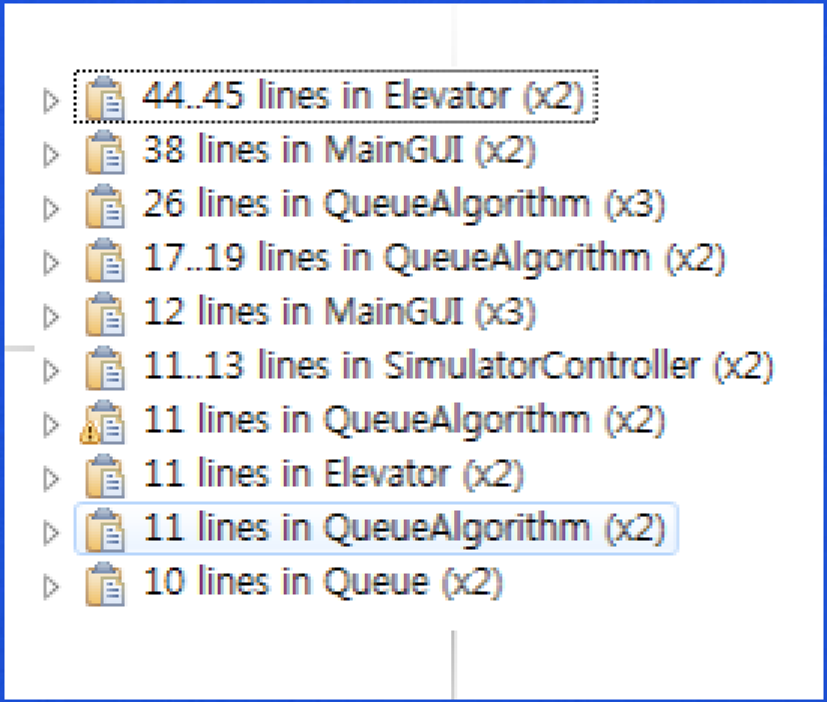
1. 0% Elevator.addLoad() : void (17)
2. 0% QueueAlgorithm.setNextDestinationByComparison(int) : void (18)
3. 0% SimulatorController.setStateRequest(int) : void (9)
4. 0% QueueAlgorithm.handleCancelRequest(int,int,int) : void (9)
5. 0% QueueAlgorithm.deQueue(QueueNode,int) : void (9)
6. 0% Elevator.MovementControl.run() : void (12)
7. 0% SimulatorController.selectLevel(int,int,int) : void (10)
8. 0% SimulatorController.setLoadRequest(int,int) : void (9)
9. 0% MainGUI.initialize(URL,ResourceBundle) : void (1)
10. 0% QueueAlgorithm.enQueue(int,int,int,int) : void (6)
11. 0% QueueAlgorithm.setEmergencyDestination(int) : void (5)
12. 0% Elevator.run() : void (5)
13. 0% SimulatorController.doorRequest(int,int) : void (6)
14. 0% QueueAlgorithm.handleSelect(int,int,int) : void (5)
15. 0% SimulatorController.requestAboard(int,int,int) : void (4)
16. 0% Elevator.setDestination(int) : void (4)
17. 0% Fluxvator.start(Stage) : void (2)
18. 0% Elevator.openDoor() : void (3)
19. 0% Elevator.closeDoor() : void (3)
20. 0% SimulatorController.cancelRequest(int,int,int) : void (5)

1. Activity, Problem and Result
CodePro : Rule

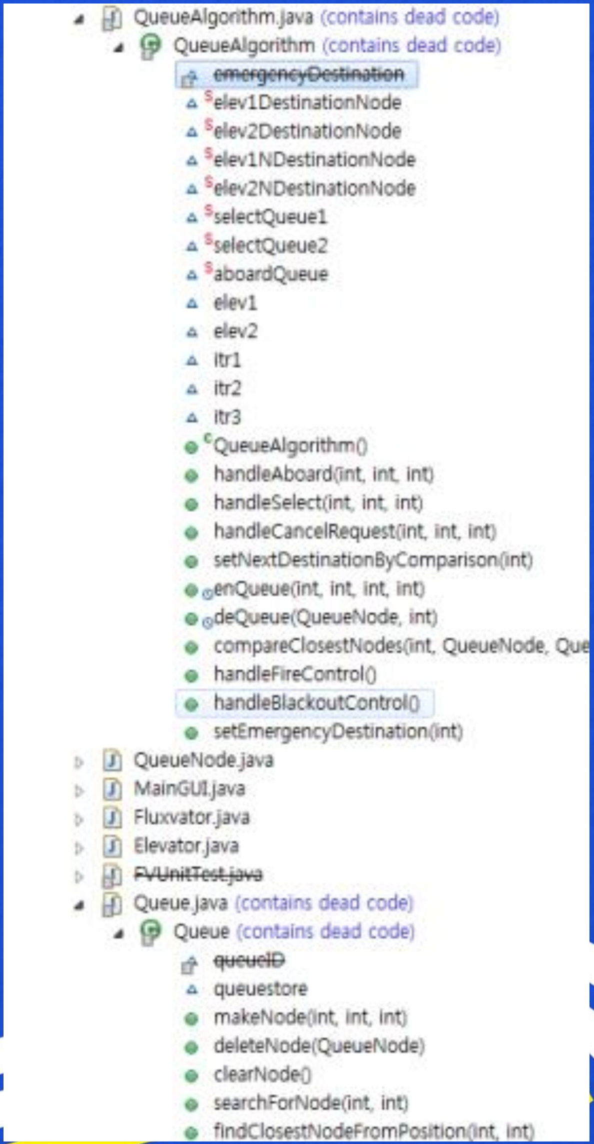


Abstractness	0%
Average Block Depth	1.01
Average Cyclomatic Complexity	3.09
Average Lines Of Code Per Method	16.39
Average Number of Constructors Per Type	0.14
Average Number of Fields Per Type	3.25
Average Number of Methods Per Type	3.22
Average Number of Parameters	0.81
Comments Ratio	3.3%
Efferent Couplings	8
Lines of Code	1,449
Number of Characters	47,690
Number of Comments	48
Number of Constructors	4
Number of Fields	88
Number of Lines	1,556
Number of Methods	87
Number of Packages	1
Number of Semicolons	696
Number of Types	27
Weighted Methods	282

1. Activity, Problem and Result CodePro : Dispersion, Duplication

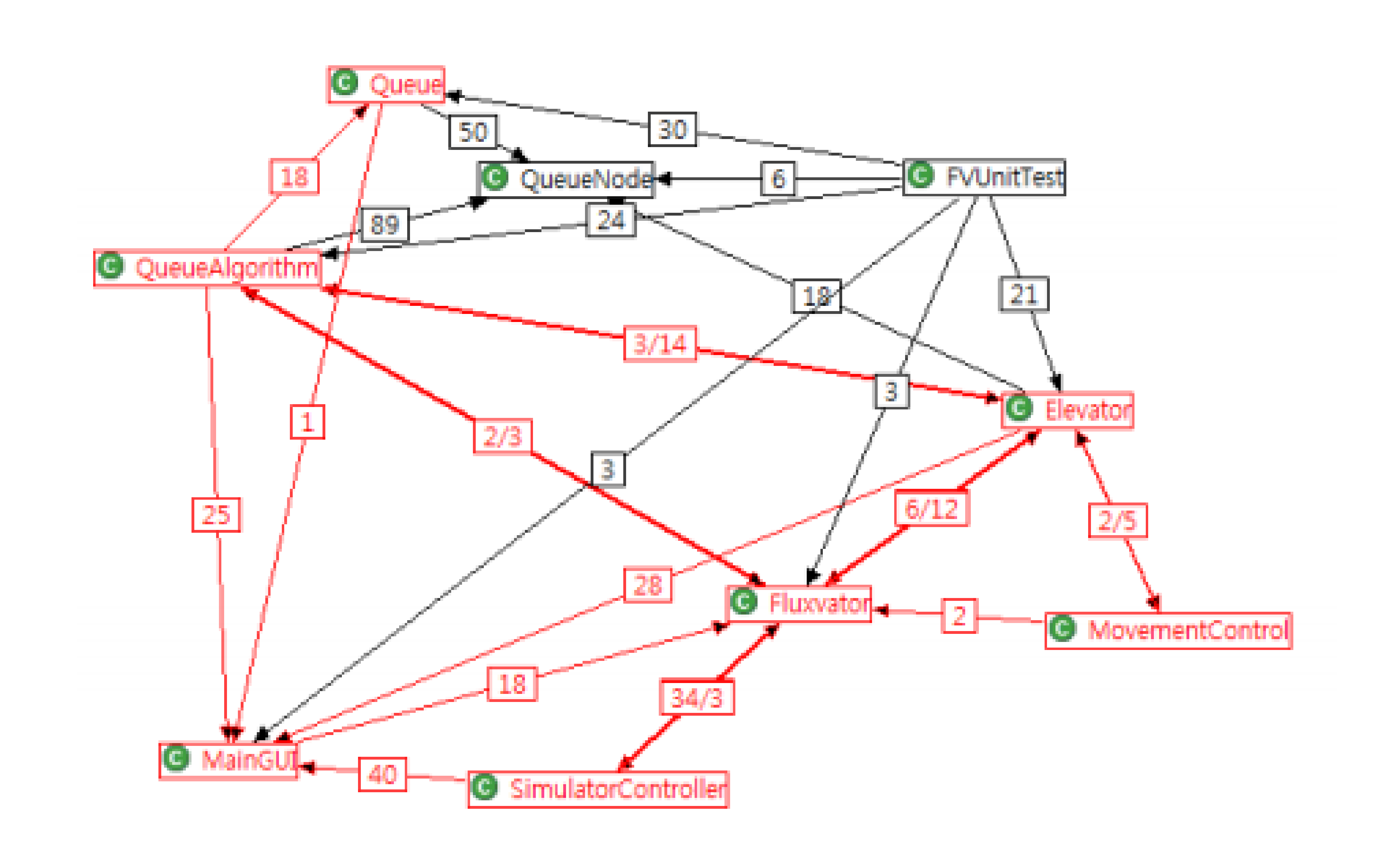


- 44..45 lines in Elevator (x2)
- 38 lines in MainGUI (x2)
- 26 lines in QueueAlgorithm (x3)
- 17..19 lines in QueueAlgorithm (x2)
- 12 lines in MainGUI (x3)
- 11..13 lines in SimulatorController (x2)
- 11 lines in QueueAlgorithm (x2)
- 11 lines in Elevator (x2)
- 11 lines in QueueAlgorithm (x2)
- 10 lines in Queue (x2)



- QueueAlgorithm.java (contains dead code)
  - QueueAlgorithm (contains dead code)
    - ~~emergencyDestination~~
    - elev1DestinationNode
    - elev2DestinationNode
    - elev1NDestinationNode
    - elev2NDestinationNode
    - selectQueue1
    - selectQueue2
    - aboardQueue
    - elev1
    - elev2
    - itr1
    - itr2
    - itr3
    - QueueAlgorithm()
    - handleAboard(int, int, int)
    - handleSelect(int, int, int)
    - handleCancelRequest(int, int, int)
    - setNextDestinationByComparison(int)
    - enQueue(int, int, int, int)
    - deQueue(QueueNode, int)
    - compareClosestNodes(int, QueueNode, Que
    - handleFireControl()
    - handleBlackoutControl()
    - setEmergencyDestination(int)
- QueueNode.java
- MainGUI.java
- Fluxvator.java
- Elevator.java
- ~~FVUnitTest.java~~
- Queue.java (contains dead code)
  - Queue (contains dead code)
    - ~~queueID~~
    - queuestore
    - makeNode(int, int, int)
    - deleteNode(QueueNode)
    - clearNode()
    - searchForNode(int, int)
    - findClosestNodeFromPosition(int, int)

1. Activity, Problem and Result
CodePro : Unused, Circulation reference,



2. What we've learned and Impression

1) Clover
    1) Too Low Coverage, meaningless Tests.
    2) Too much importance on specific class

2) Code Pro
    1) Static Luins Everything
    2) Class Dependency
    3) Memory Leak!(Thread still Alive!)

Impression of Software Verification

Sangyoon Kim :

자신이 정말 개발자의 길을 걷고자 한다면 반드시 수강해야 합니다. 진짜로. System에 대한 분석하는 방법 뿐만 아니라, 그 과정에서 산출된 결과물에 대해 스스로의 코딩 방식과 대조해 보는 의미 있는 수업을 들었다고 자신할 수 있습니다.

그리고 역시 자바는 할게 못됩니다.

Green Kim :

CTIP환경을 구성하고 이 환경 내에서 프로젝트를 진행해 본 경험이 없기에 처음엔 생소하게 느껴진 게 사실이었습니다. 하지만 프로젝트를 진행하며 이 환경 내에서 지속적으로 프로그램을 관리하고 수행하며 중요성과 한계점에 대해 배울 수 있는 시간이었습니다. 더불어 system testing과 static analysis를 통해 프로그램을 어떻게 개발하고 더 나아가 어떻게 검증 해야 하는가에 대한 청사진을 마련할 수 있는 뜻 깊은 시간이었습니다.

무엇보다 개발자로서 knowledge와 experience를 쌓는 다는 것이 얼마나 중요한지 절실히 깨달을 수 있었습니다.

Thank You

Q & A