

Fundamentals of Test Processes



- 1st Korean SPICE Network Workshop
10th, December, 2019, Seoul -

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

- Test level : DO

2. BUG

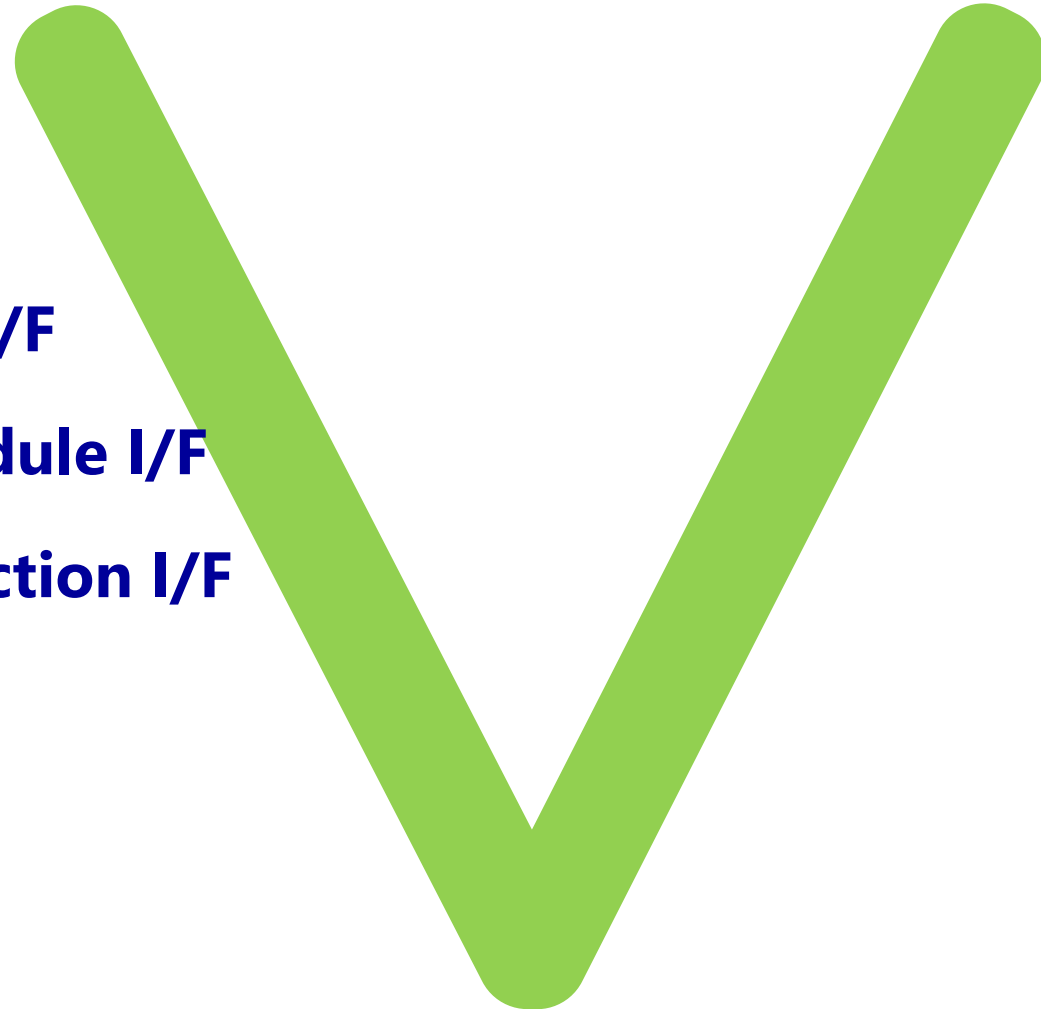
- Biggest problem

- Complete bug fix

1. External I/F

2. Inter-module I/F

3. Inter-function I/F



TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

- Test level : DO

2. BUG

- Biggest problem

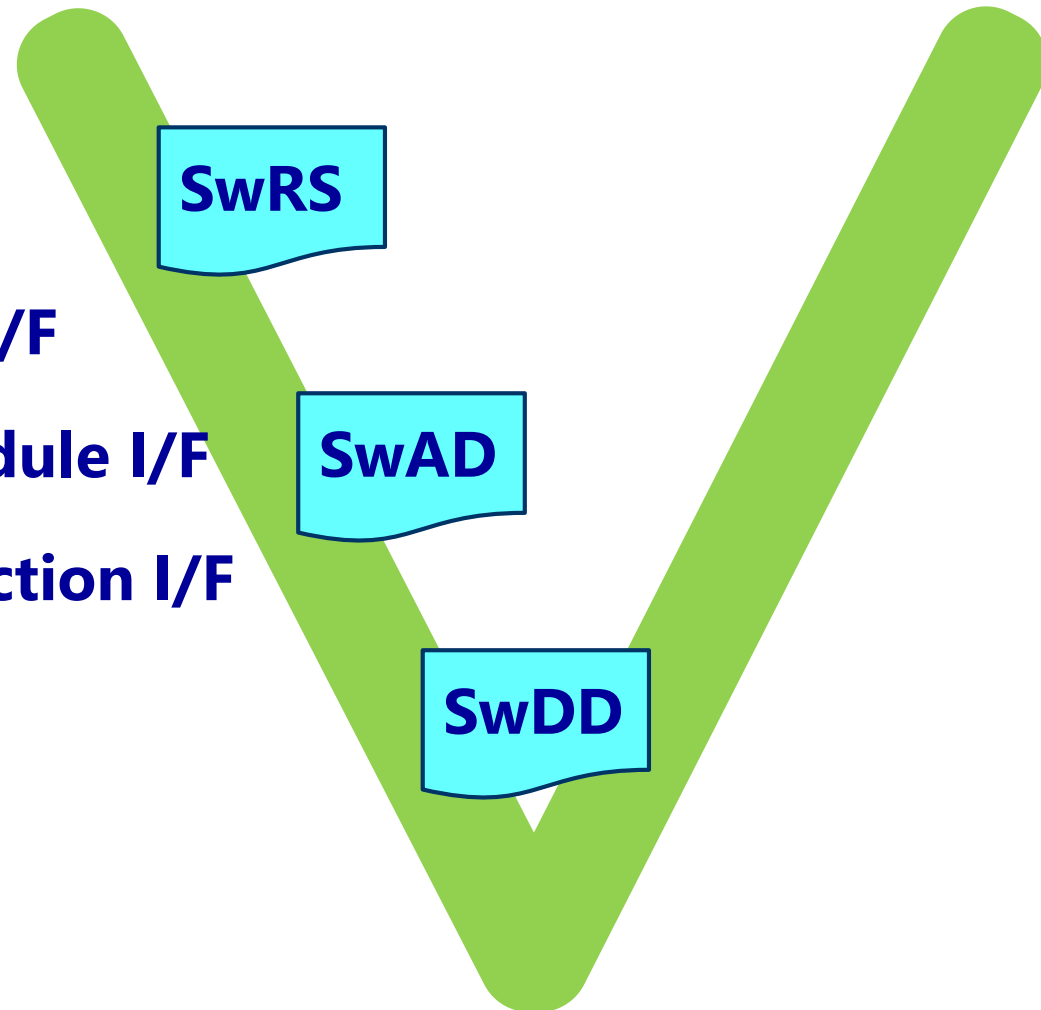
- Complete bug fix

[CASE #1]

1. External I/F

2. Inter-module I/F

3. Inter-function I/F



TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

- Test level : DO

2. BUG

- Biggest problem

- Complete bug fix

[CASE #2]

1. External I/F

SwRS

2. Inter-module I/F

SwAD

3. Inter-function I/F

SwDD

TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

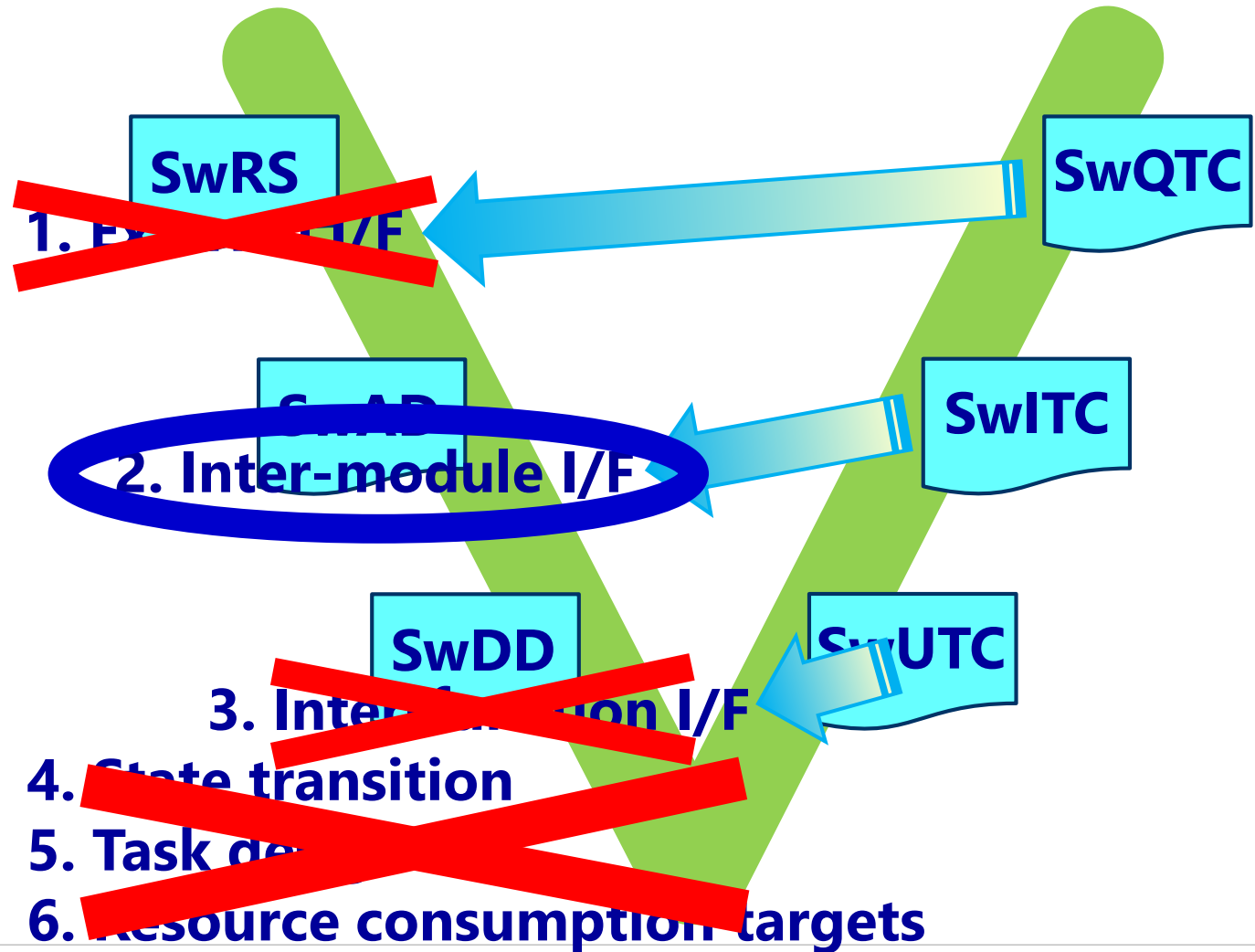
- Test level : DO

2. BUG

- Biggest problem

- Complete bug fix

[CASE #2]



■ TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

- Test level : DO

2. BUG

- Biggest problem

- Complete bug fix

[CASE #2]

0. Requirements

1. External I/F

2. Inter-module I/F

3. Inter-function I/F

4. State transition

5. Task design

6. Resource consumption targets

7. Algorithm design

etc...

■ TEST STRATEGY - Biggest problem

INDEX

1. TEST STRATEGY

- Biggest problem

- Test level : IDEA

- Test level : DO

2. BUG

- Biggest problem

- Complete bug fix

(IRON RULE)
Every single
specification in the
LEFT side of V has
to be verified by
the **RIGHT** side of V

■ TEST STRATEGY - Test level : IDEA > Quantity

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- > Quantity
- ~ 2nd level scope
- > Quality

- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

[2nd LEVEL of TEST SCOPE]

For example ;

“test all requirements?”

VS

“test only changed parts?”

-> QUANTITY

■ TEST STRATEGY - Test level : IDEA (CONCEPT)

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - > Quality
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

TEST LEVEL : 2 dimensions

1. QUANTITY

2. QUALITY

■ TEST STRATEGY - Test level : IDEA > Quality

INDEX

1. TEST STRATEGY

- Biggest problem
- **Test level : IDEA**
 - > Quantity
 - + 2nd lv test scope
 - > **Quality**
 - + **Test cases**
 - + **Pass criteria**
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

QUALITY = HOW STRICT TEST?

QUALITY : 2 dimensions

HOW STRICT?

1. VERIFICATION CRITERIA

= TEST CASES

2. PASS CRITERIA

■ TEST STRATEGY - Test level : IDEA

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

3 QUESTIONS

1. 2nd level of test scope
2. How strict test cases
3. How strict pass criteria

TEST STRATEGY - Test level : IDEA

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

4 CASES

	PROTOTYPE	PreMP (양산직전)
NO CHANGE	1	2
CHANGED	3	4

TEST LEVEL : IDEA ~ 1st QUESTION

INDEX


1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- > Quantity
- + 2nd lv test scope
- > Quality
- + Test cases
- + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

1. 2nd level of test scope
2. How strict test cases
3. How strict pass criteria

	PROTOTYPE	PreMP (양산직전)
NO CHANGE	1	2
CHANGED	3	4

TEST LEVEL : IDEA ~ 2nd QUESTION

INDEX

1. TEST STRATEGY

- Biggest problem
- **Test level : IDEA**
 - > Quantity
 - + 2nd lv test scope
 - > **Quality**
 - + **Test cases**
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

1. 2nd level of test scope
2. **How strict test cases**
3. How strict pass criteria

Error cases needed?	PROTOTYPE	PreMP (양산직전)
NO CHANGE	1	2
CHANGED	3	4

TEST LEVEL : IDEA ~ 3rd QUESTION

INDEX

1. TEST STRATEGY

- Biggest problem
- **Test level : IDEA**
 - > Quantity
 - + 2nd lv test scope
 - > **Quality**
 - + Test cases
 - + **Pass criteria**

- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

1. 2nd level of test scope
2. How strict test cases
3. How strict pass criteria



	PROTOTYPE	PreMP (양산직전)
NO CHANGE	1	2
CHANGED	3	4

■ TEST LEVEL : IDEA ~ CONCLUSION

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : **IDEA**
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

MAX SCOPE is not always necessary, MAX STRICT is not always necessary.

...But, is it OK with Automotive SPICE?

■ TEST LEVEL : IDEA ~ CONCLUSION

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

**ANSWER ;
It's OK.**

**But only if
VALID RATIONALE**

■ TEST LEVEL : IDEA ~ CONCLUSION

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

How can we derive
VALID RATIONALE?
= STRATEGY

■ TEST LEVEL : IDEA ~ STRATEGY

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
 - > Quantity
 - + 2nd lv test scope
 - > Quality
 - + Test cases
 - + Pass criteria
- Test level : DO

2. BUG

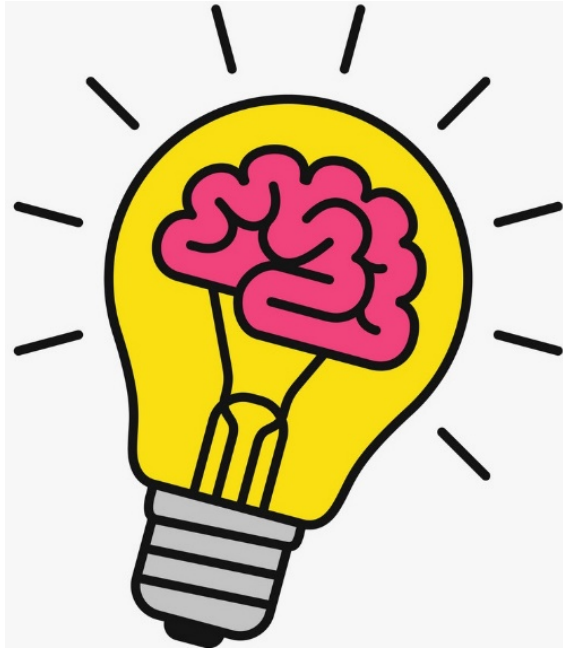
- Biggest problem
- Complete bug fix

**6 supporting &
all 5 test processes
require STRATEGY**

WHY?

**Because not always
necessary**

TEST LEVEL : IDEA ~ STRATEGY



- ◆ **Project with Only Geniuses...**
- ◆ **Needs no check, no review, no test, because they are perfect!!!**
 - SUP.1
 - SUP.2
 - SWE.4~6, SYS.4, SYS.5
 - SUP.9
- ◆ **All the documents are always perfect...**
 - SUP.8
 - SUP.10

TEST LEVEL : IDEA ~ STRATEGY



- SUP.1
- SUP.2
- SWE.4~6, SYS.4, SYS.5
- SUP.9
- SUP.8
- SUP.10

◆ A Real Project ...

◆ Needs all of them ~ review, check, test, baseline, etc.

◆ But wait a second, doesn't it depend on the situation?

- From prototyping, do you really need to test meticulous error cases?
- For preMP, can you really release it without testing any error case?
- Do you need to review 20 years expert as strictly as freshman?

■ TEST LEVEL : IDEA ~ STRATEGY

INDEX

1. TEST STRATEGY

- Biggest problem
- **Test level : IDEA**
 - > Quantity
 - + **2nd lv test scope**
 - > Quality
 - + **Test cases**
 - + **Pass criteria**

2. BUG

**“GOOD LEVEL”
is always different
depending on
the situation**

**How can we derive
this GOOD LEVEL?**

■ TEST LEVEL : DO

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- **Test level : DO**
 - > Necessity level
 - > Recipe

2. BUG

2 STEPs for TEST STRATEGY

Identify

1. Necessity level
2. Recipe

■ TEST LEVEL : DO > Necessity Level

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- **Test level : DO**
 - > Necessity level**
 - + how important?**
 - + how risky?**
- > Recipe

2. BUG

2 CRITERIA

1. How important?
2. How risky?

■ TEST LEVEL : DO > Necessity Level

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- **Test level : DO**
- > Necessity level**
- + how important?**
- + how risky?
- > Recipe

2. BUG

HOW IMPORTANT?

For example ;

1. Features ; main? sub?
2. Release ; prototype? preMP?
3. Project rank ; high? low?

■ TEST LEVEL : DO > Necessity Level

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- **Test level : DO**
- > Necessity level**
- + how important?
- + **how risky?**
- > Recipe

2. BUG

HOW RISKY?

For example ;

1. Changes ; many? little?
2. Complexity ; difficult? easy?
3. Skill ; freshman? expert?
4. ASIL ; high? low?

■ TEST LEVEL : DO > Recipe

INDEX

1. TEST STRATEGY

- Biggest problem
 - Test level : IDEA
 - **Test level : DO**
 - > Necessity level
 - > **Recipe**
 - + **IDEA**
 - ~ **Scope**
 - ~ **Test case**
 - ~ **Pass criteria**
 - + **DO**
- ### 2. BUG

ADJUST (COOK) TEST LEVEL
according to the above
NECESSITY LEVEL

3 SPICES for RECIPE ;

- 1. Scope**
- 2. Test case**
- 3. Pass criteria**

■ TEST LEVEL : DO > Recipe ~ SCOPE

INDEX

1. TEST STRATEGY

- Biggest problem
 - Test level : IDEA
 - **Test level : DO**
 - > Necessity level
 - > **Recipe**
 - + **IDEA**
 - ~ **Scope**
 - ~ Test case
 - ~ Pass criteria
 - + **DO**
- ### 2. BUG

For example ;

- **Change vs No change**
- **Main feature vs Sub**
- **Difficult feature vs Easy**

■ TEST LEVEL : DO > Recipe ~ TEST CASE

INDEX

1. TEST STRATEGY

- Biggest problem
 - Test level : IDEA
 - **Test level : DO**
 - > Necessity level
 - > **Recipe**
 - + **IDEA**
 - ~ Scope
 - ~ **Test case**
 - ~ Pass criteria
 - + DO
- ### 2. BUG

For example ;

- **Regular cases vs Error cases**
- **ref. ISO26262**
 - > **analysis of requirements**
 - > **equivalence partitioning**
 - > **boundary value analysis**
 - > **error guessing**

■ TEST LEVEL : DO > Recipe ~ PASS CRITERIA

INDEX

1. TEST STRATEGY

- Biggest problem
 - Test level : IDEA
 - **Test level : DO**
 - > Necessity level
 - > **Recipe**
 - + **IDEA**
 - ~ Scope
 - ~ Test case
 - ~ **Pass criteria**
 - + DO
- ### 2. BUG

Define bug severity level

For example ;

1. Critical
2. Major
3. Minor
4. Observatory

■ TEST LEVEL : DO > Recipe : DO

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- **Test level : DO**
- > Necessity level
- > **Recipe**
- + IDEA
- + **DO**
- ~ **Prototype**
- ~ **100% feature**
- ~ **Bug fix**
- ~ **PreMP**

2. BUG

4 CASES (PHASES)

1. **Prototype**
2. **100% feature**
3. **Bug fix**
4. **PreMP**

TEST LEVEL : DO > Recipe : DO

	Proto type	100% feature	Bug fix	PreMP
Scope	Only changed & main features	All changed (main / sub) features	All changed, no changed & main features	All features
Test case	Only regular cases	Equivalence partitioning	Boundary value analysis	Error guessing
Pass criteria	Critical 0% Major 3% Minor 10% Obsv. 25%	Critical 0% Major 0% Minor 3% Obsv. 10%	Critical 0% Major 0% Minor 0% Obsv. 5%	Critical 0% Major 0% Minor 0% Obsv. 2%

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

BUG : Biggest problem ~ Incomplete bug fix

INDEX

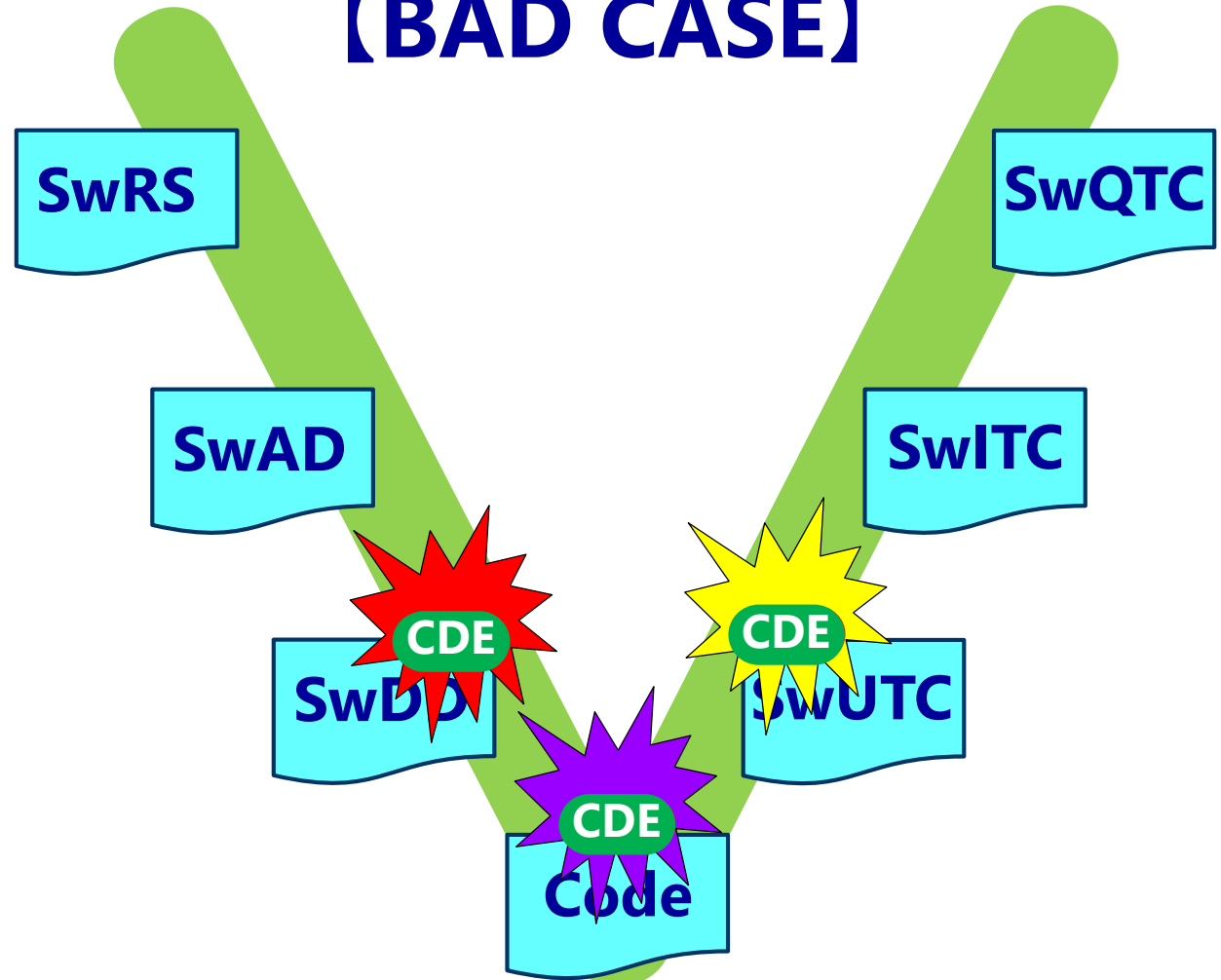
1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

(BAD CASE)



BUG : Biggest problem ~ Incomplete bug fix

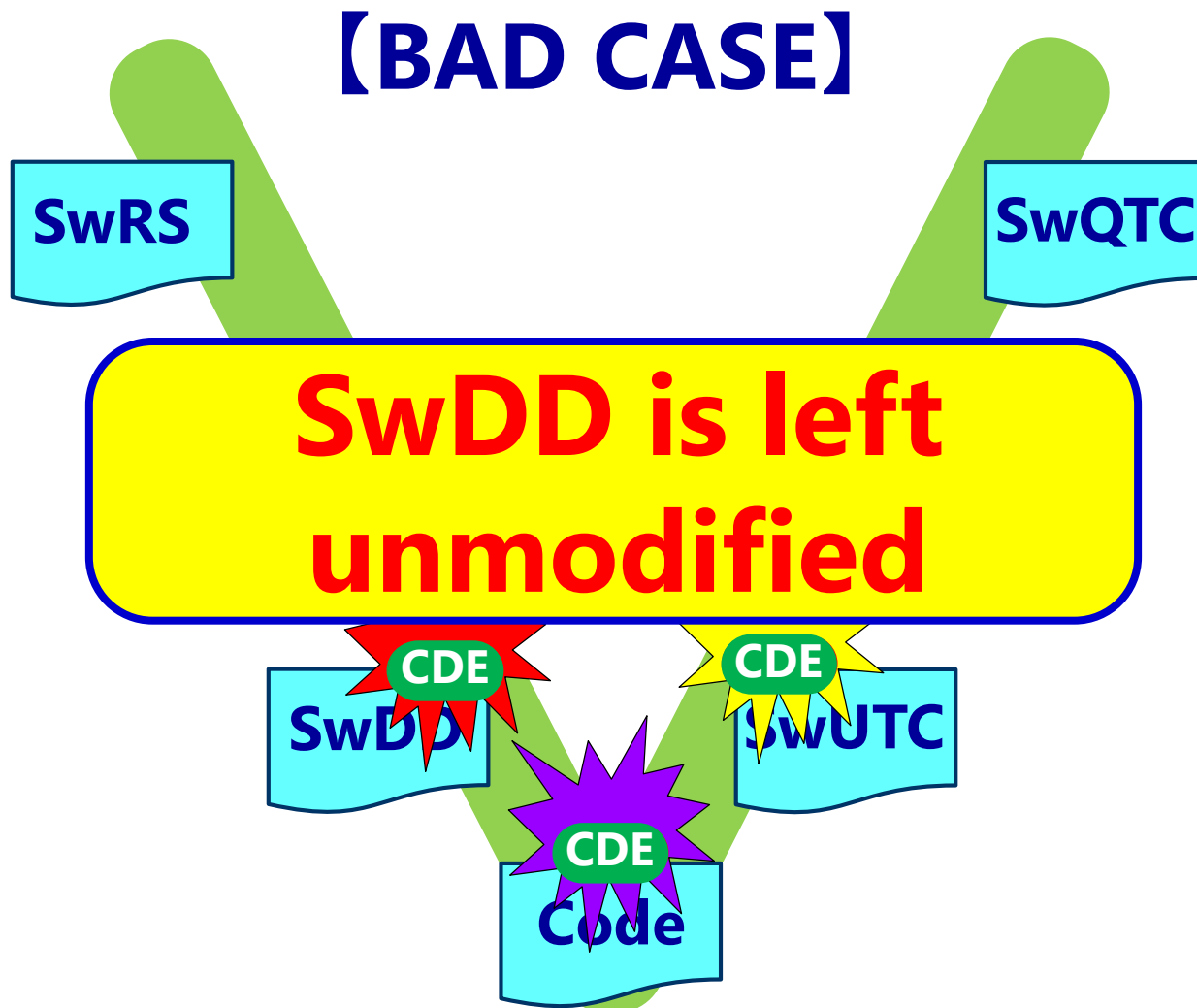
INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix



BUG : Biggest problem ~ Incomplete bug fix

INDEX

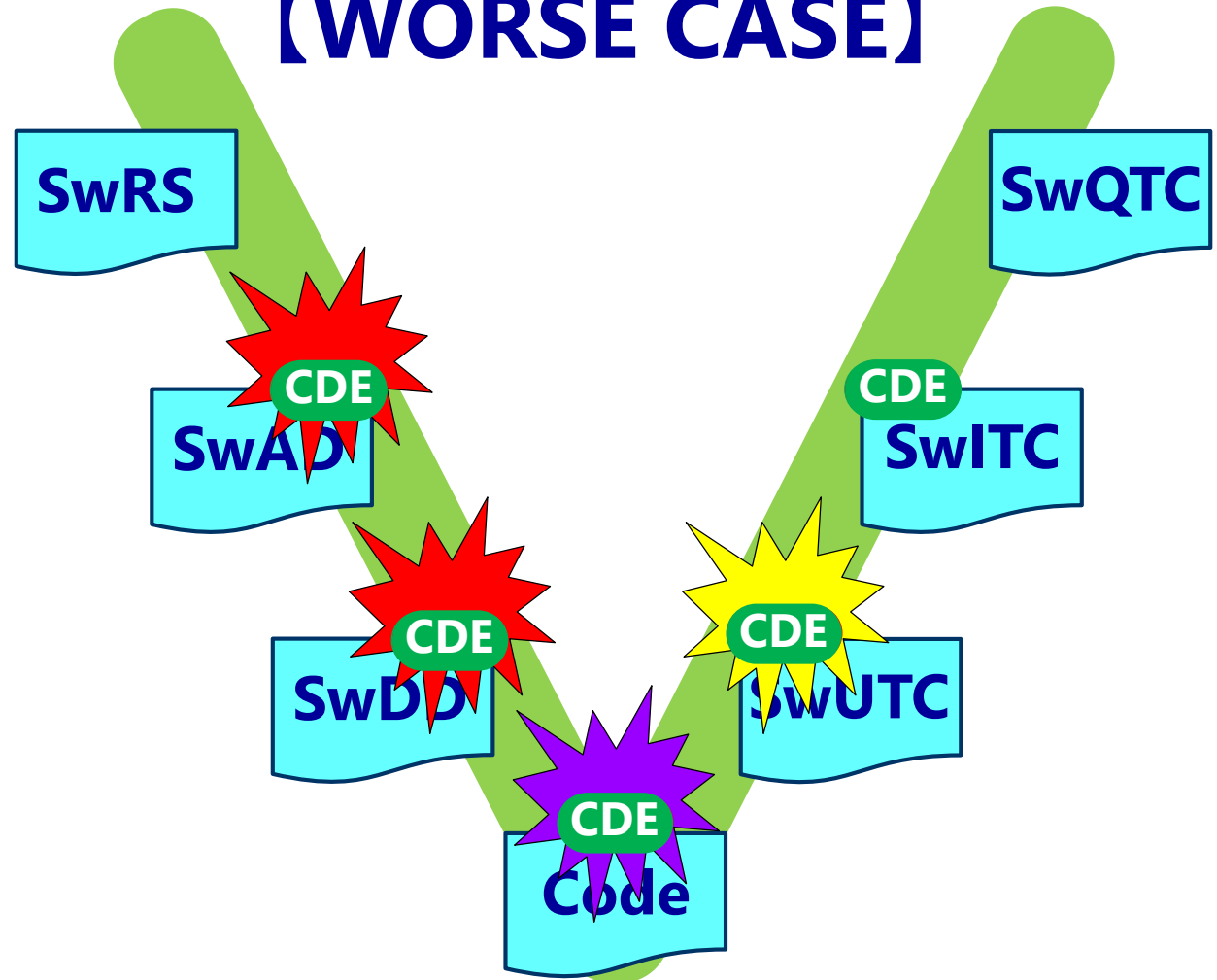
1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- **Biggest problem**
- Complete bug fix

(WORSE CASE)



■ BUG : Biggest problem ~ Incomplete bug fix

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

Inconsistency among
1.Engineering document
2.Source codes
and 3.Test cases

Consistency among 3 is
primary entry criteria
for test execution

BUG : Biggest problem ~ Incomplete bug fix

INDEX

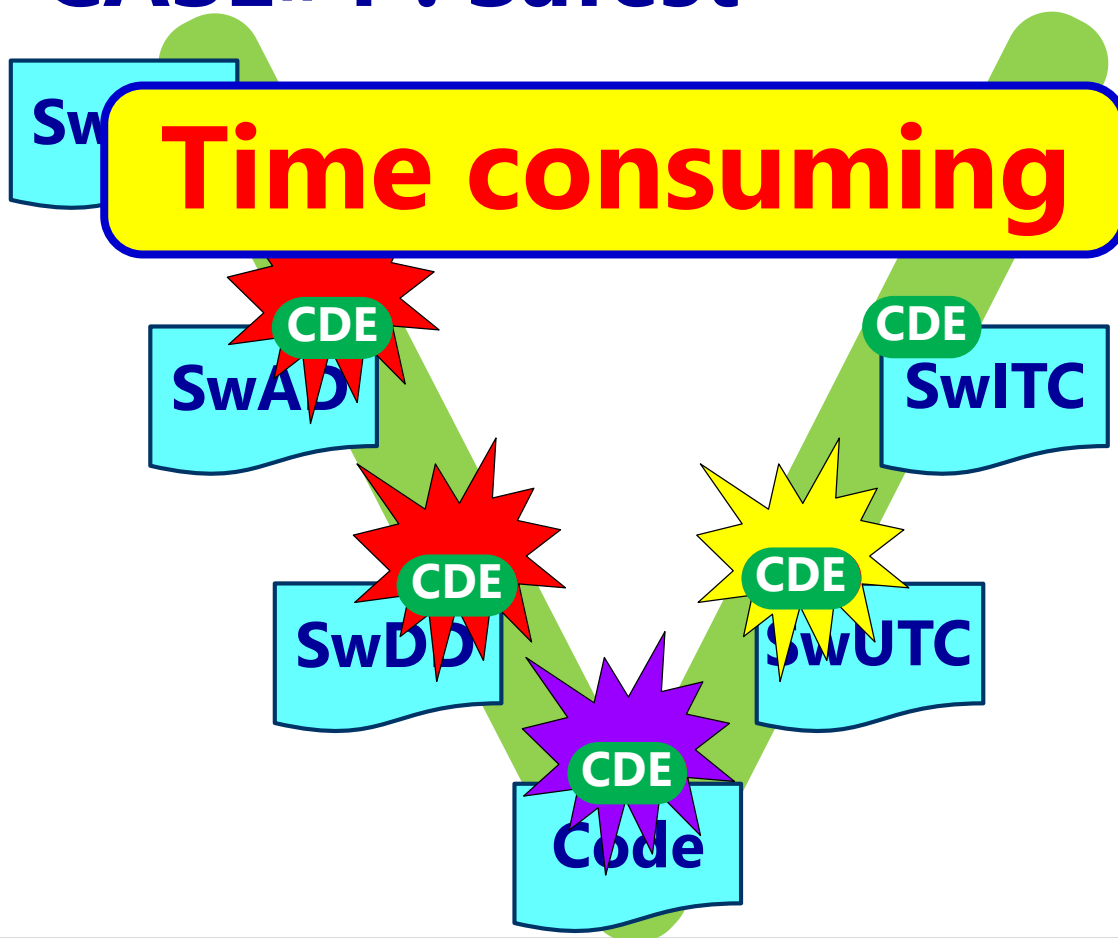
1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

How to ensure consistency? ~ CASE#1 : Safest



BUG : Biggest problem ~ Incomplete bug fix

INDEX

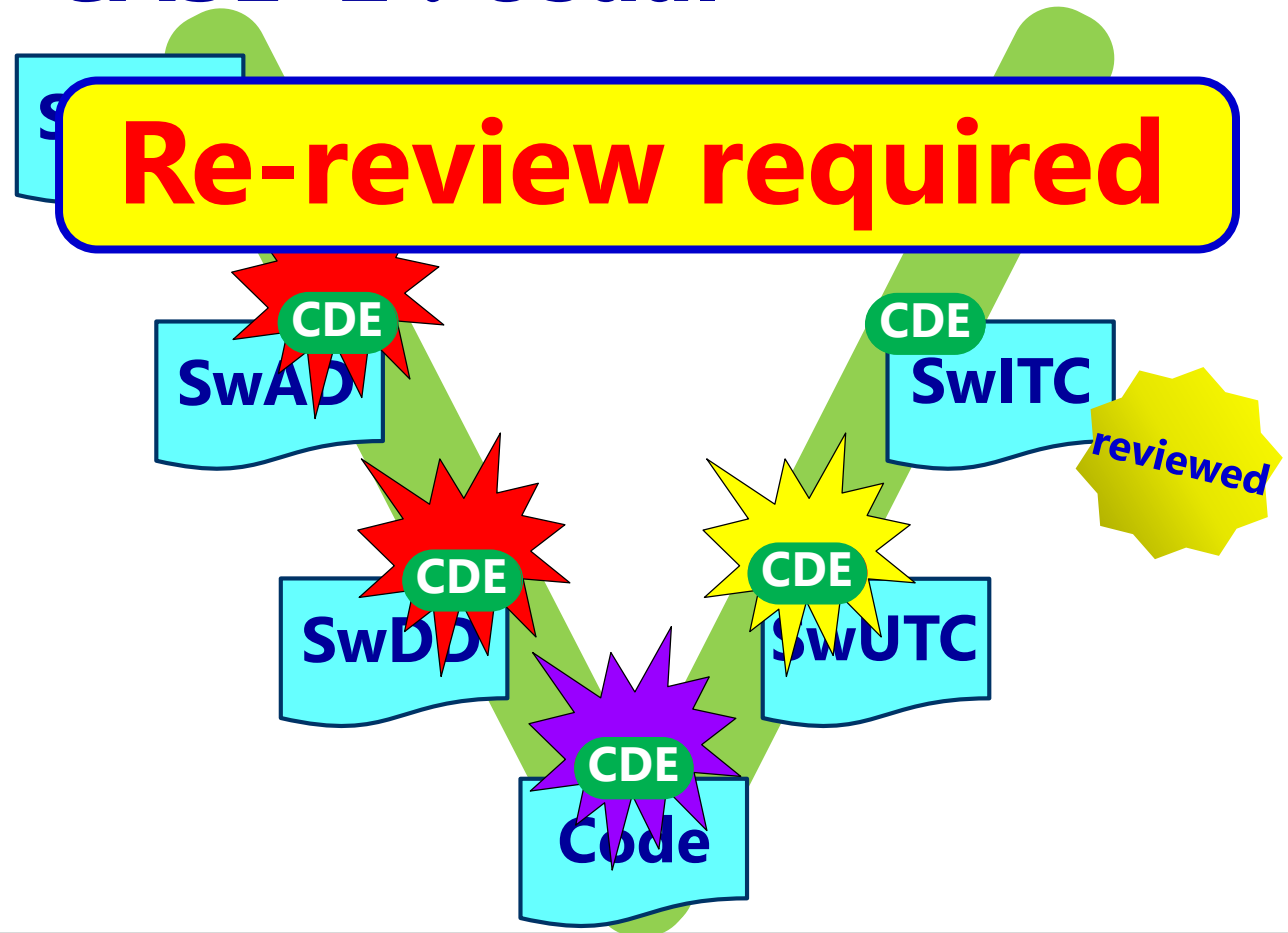
1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- **Biggest problem**
- Complete bug fix

How to ensure consistency? ~ CASE#2 : Usual



BUG : Biggest problem ~ Incomplete bug fix

INDEX

1. TEST STRATEGY

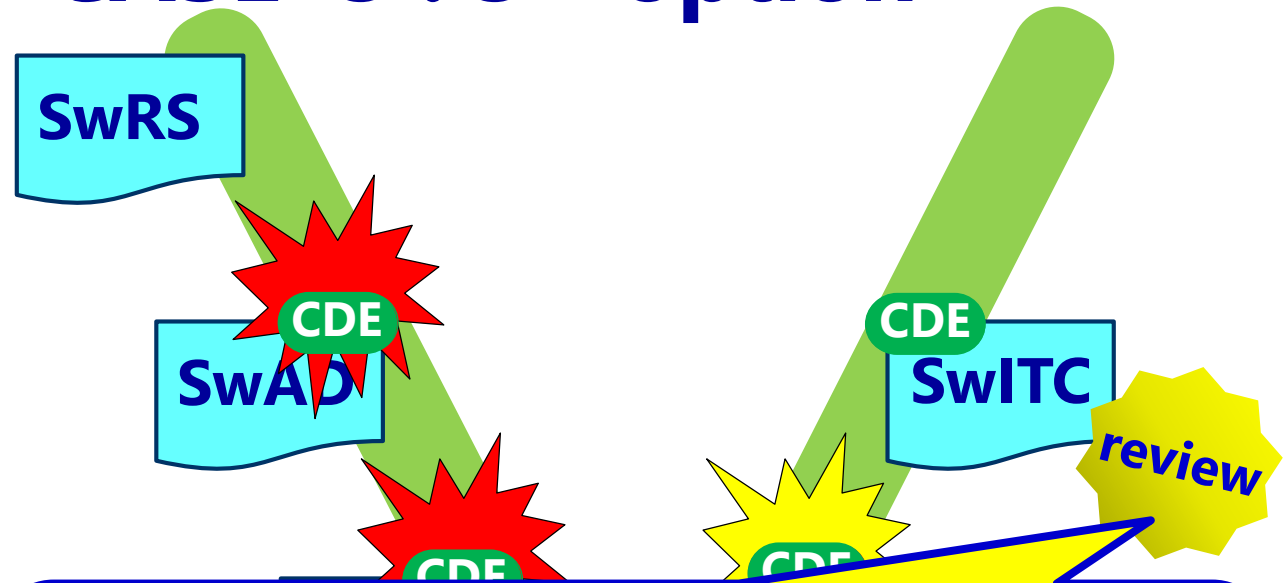
- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix

How to ensure consistency?

~ CASE#3 : 3rd option



This review checks consistency among 3

■ BUG : Biggest problem ~ Incomplete bug fix

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- **Biggest problem**
- Complete bug fix

4 ENTRY CRITERIA for TEST

1. Test plan development
2. Test case development
3. Test case review
4. Test execution

■ BUG : Complete bug fix

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- **Complete bug fix**
 - > **Upstream**
 - > **Downstream**
 - > **Horizontal**

Complete bug fix

≡ **REGRESSION TEST STRATEGY**

Consists of **3 STEPs**

1. UPSTREAM

2. DOWNSTREAM

3. HORIZONTAL

BUG : Complete bug fix

INDEX

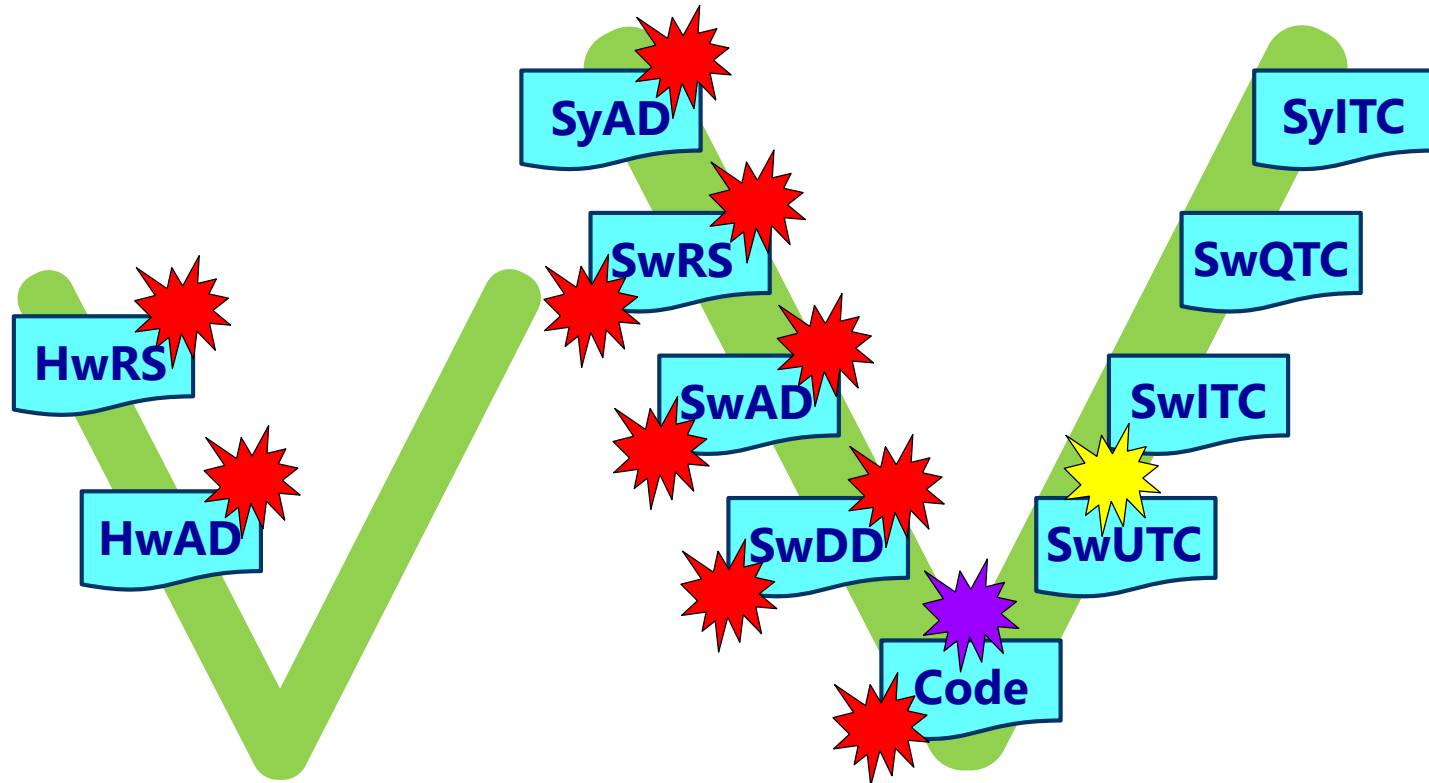
1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- **Complete bug fix**
 - > Upstream
 - > Downstream
 - > Horizontal

(FURTHER WORSE CASE)



BUG : Complete bug fix

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- Complete bug fix
 - > Upstream
 - > Downstream
 - > Horizontal

[1st STEP] [2nd STEP]

**1st step : UPSTREAM
DEEPEST ROOT CAUSE**



**2nd step : DOWNSTREAM
CHANGE IMPACT SCOPE
on LEFT side of V**

■ BUG : Complete bug fix ~ 3rd HORIZONTAL

INDEX

1. TEST STRATEGY

- Biggest problem
- Test level : IDEA
- Test level : DO

2. BUG

- Biggest problem
- **Complete bug fix**
 - > Upstream
 - > Downstream
 - > **Horizontal**

**3rd step : HORIZONTAL
RETEST SCOPE
on RIGHT side of V**

**However,
CHANGE SCOPE (left)
≠ RETEST SCOPE (right)**

SwRS

SwQTC

**Thank you very much
for your attention!**

Please feel free to contact us if any questions



jkkim@cnbis.co.kr