

Software Verification Summary

; 2018 Version

201411259 교수창
201412005 이세라
201411314 전소영
201511304 하지윤

INDEX

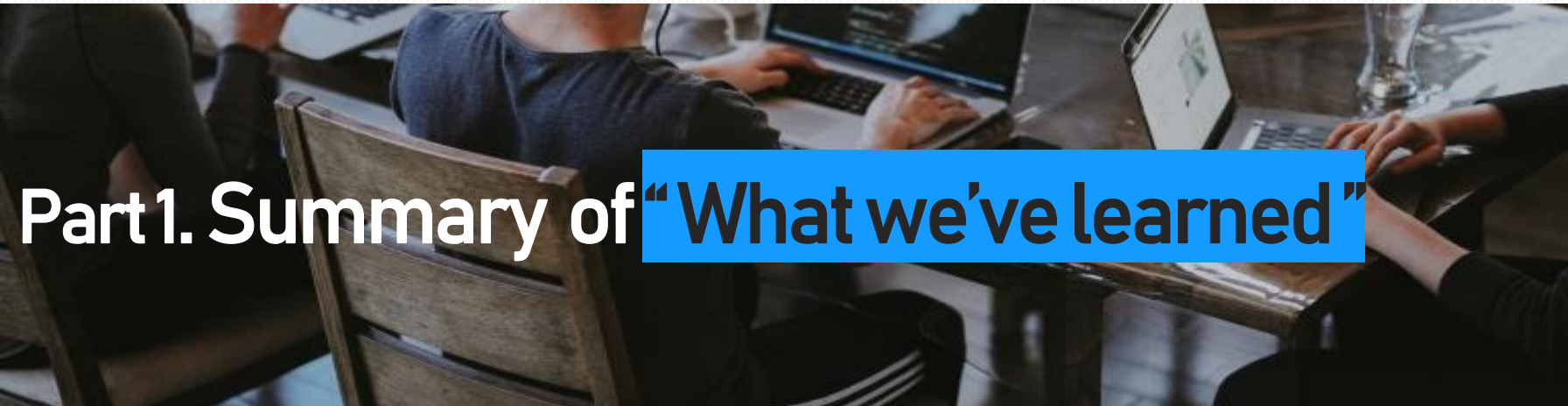
Part 1. Summary of What we've learned

- 01. Theoretical Aspects
- 02. Our CTIP & Uses
- 03. Testing with SM Team

Part 2. Lessons from Software Verification

- 01. When construct CTIP
- 02. When Testing with SM Team

Part 3. Suggestion for Future Lecture



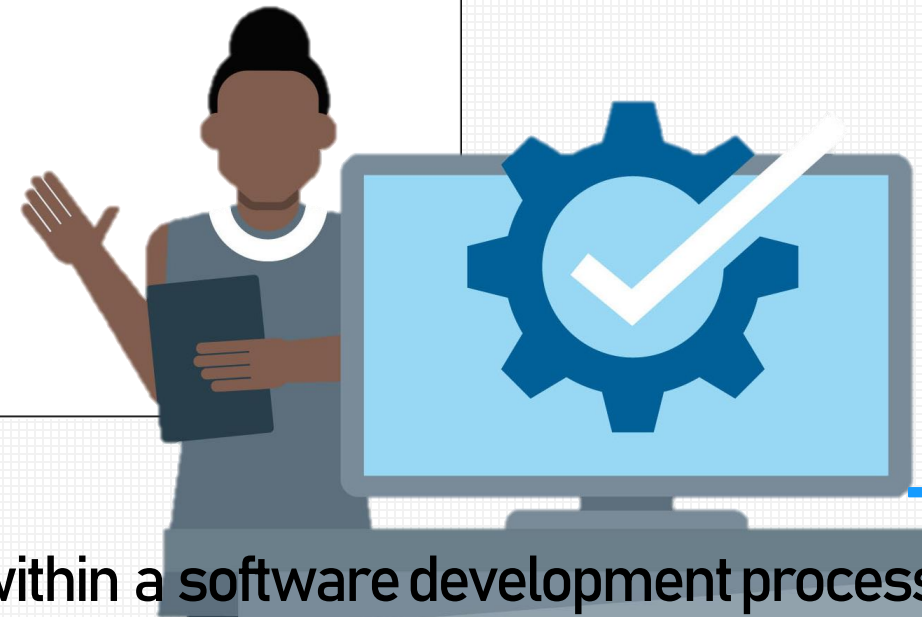
Part 1. Summary of "What we've learned"

- 01. Theoretical Aspects
 - 02. Our CTIP & Uses
 - 03. Testing with SM Team
-

01. Theoretical Aspects

; What was the Ultimate goal of this lecture?

Fundamentals of Test&Analysis	Chapter 1. Software Test and Analysis in a Nutshell Chapter 2. A Framework for Test and Analysis Chapter 3. Basic Principles Chapter 4. Test and Analysis Activities Within a Software process
Basic Techniques	Chapter 5. Finite Models Chapter 6. Dependence and Data Flow Models Chapter 8. Finite State Verification Chapter 9. Test Case Selection and Adequacy
Problems and Methods	Chapter 10. Functional Testing Chapter 11. Combinatorial Testing Chapter 12. Structural Testing Chapter 13. Data Flow Testing Chapter 14. Model based Testing Chapter 16. Fault based Testing Chapter 17. Test Execution Chapter 19. Program Analysis



Provide the rationale for selecting and combining them within a software development process

02. Our CTIP & Uses

; Which part was most useful during the project?

Build



CI



Jenkins

Issue/Code Managing



GitHub

System Test



TestLink

Static Analysis



FindBugs



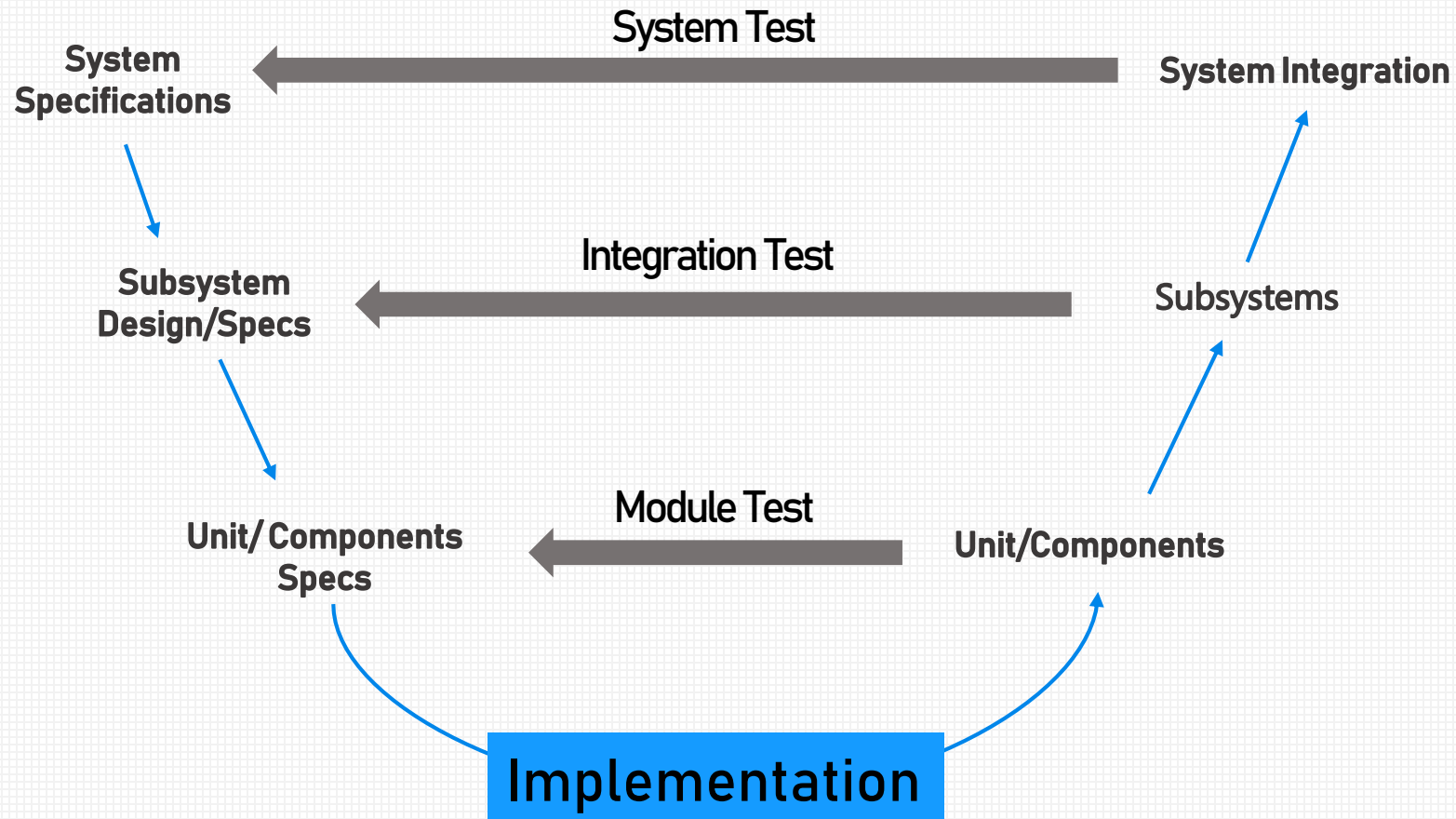
CheckStyle



PMD

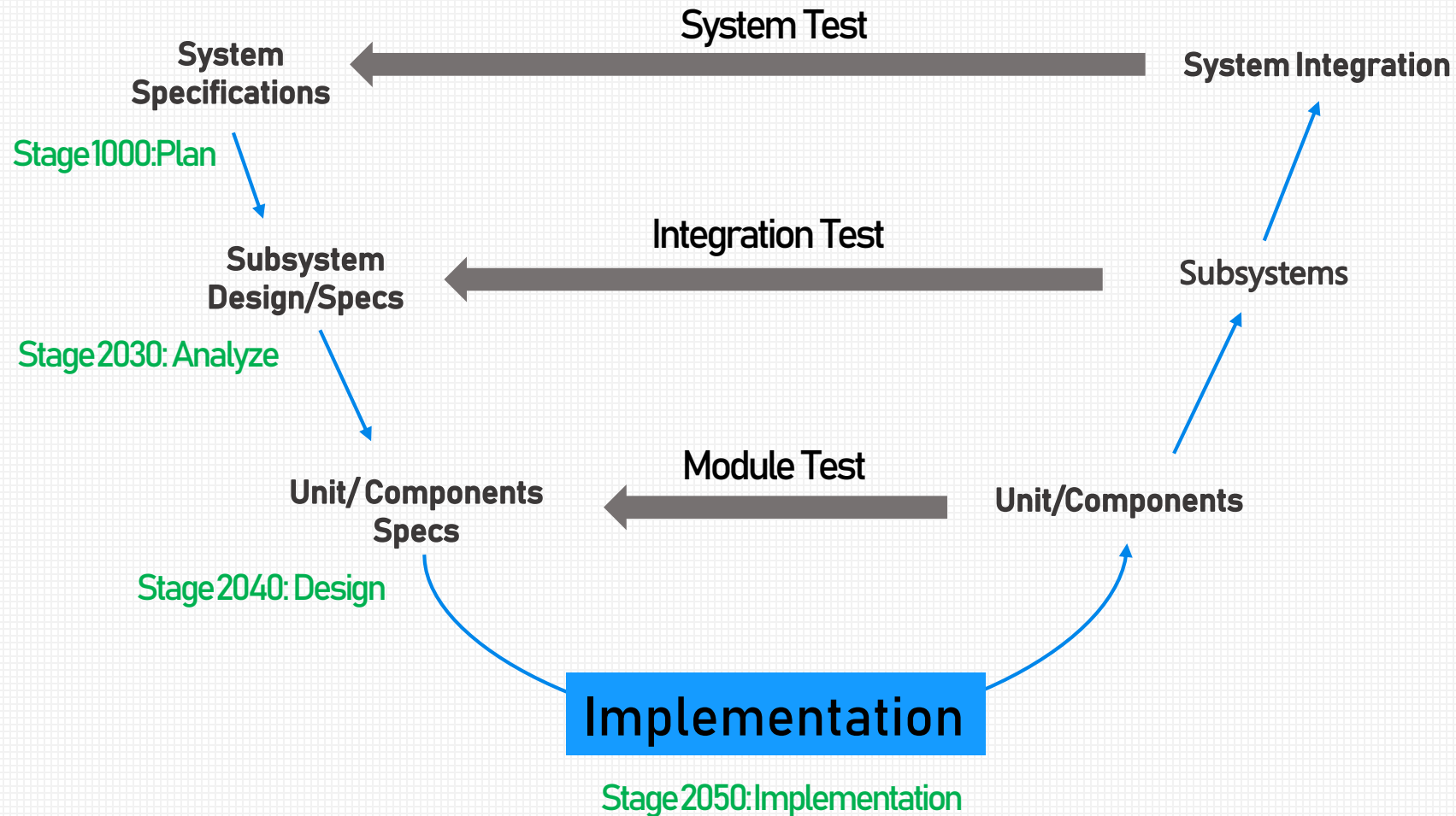
03. Testing with SM Team

; Did we follow the V&V model while doing a project?



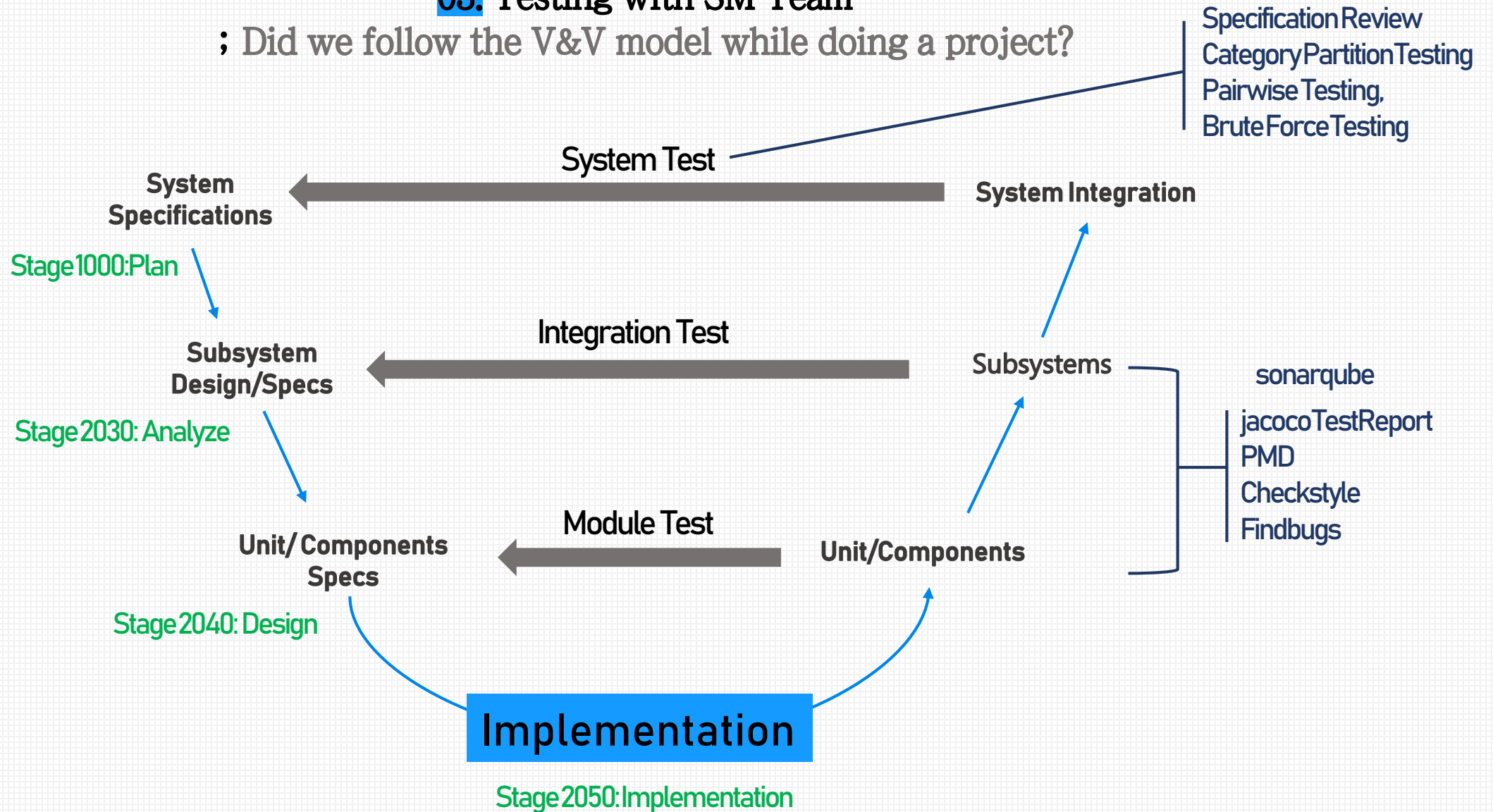
03. Testing with SM Team

; Did we follow the V&V model while doing a project?



03. Testing with SM Team

; Did we follow the V&V model while doing a project?





Part 2. Lessons from “Software Verification”

01. When construct CTIP

02. When Testing with SM Team

- CPT and Static Analysis
 - Thoughts about Team project
-

01. When Construct CTIP

; In different perspective, what's the pros and cons of using CTIP?

		as a Software Developer	as a Software Quality Engineer
Pros	SV	Convenient Build	Automatically verified code
		Offer Convenience by removing some static error	Convenient to check result
	SM	Helpful not to forgetting the issues and problems	
		Make easy to Cooperate with team	
		Can fix the bad habits when coding	
Cons	SV	Misuse of rules and tools can cause more error and warning (More Stress)	Be aware that tools are not complete
			Analyze outcome ability is positively necessary
		Need to know how to use tools	Maintenance cost of Server
			Using right amount of tools (More doesn't mean better)

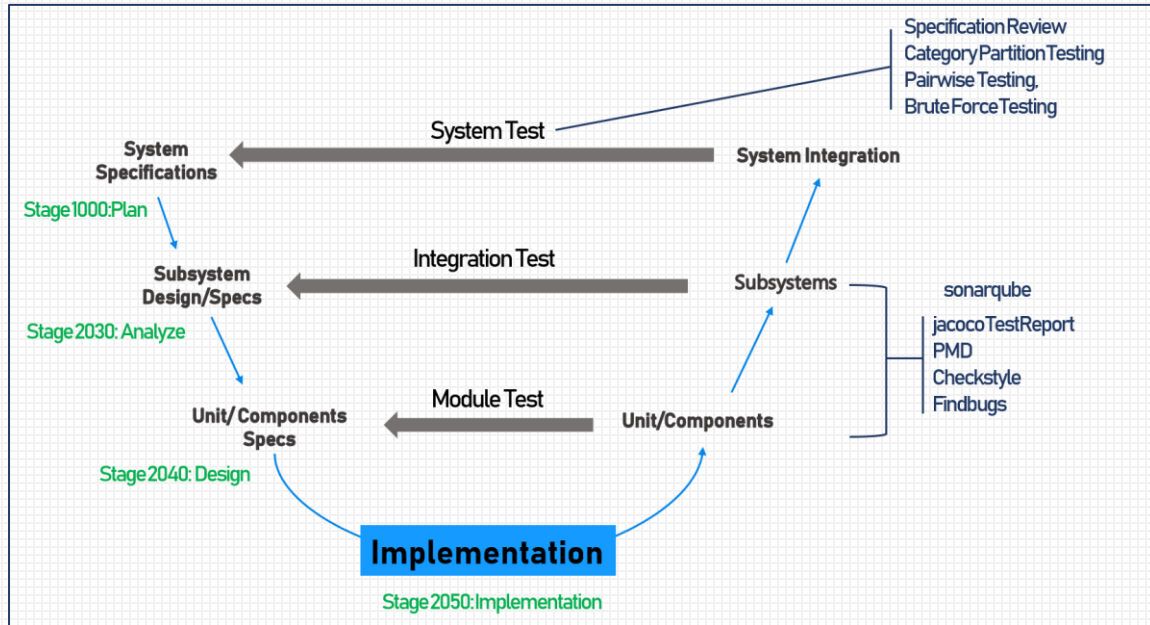
“

It was most Realistic lecture in the entire Computer major curriculums

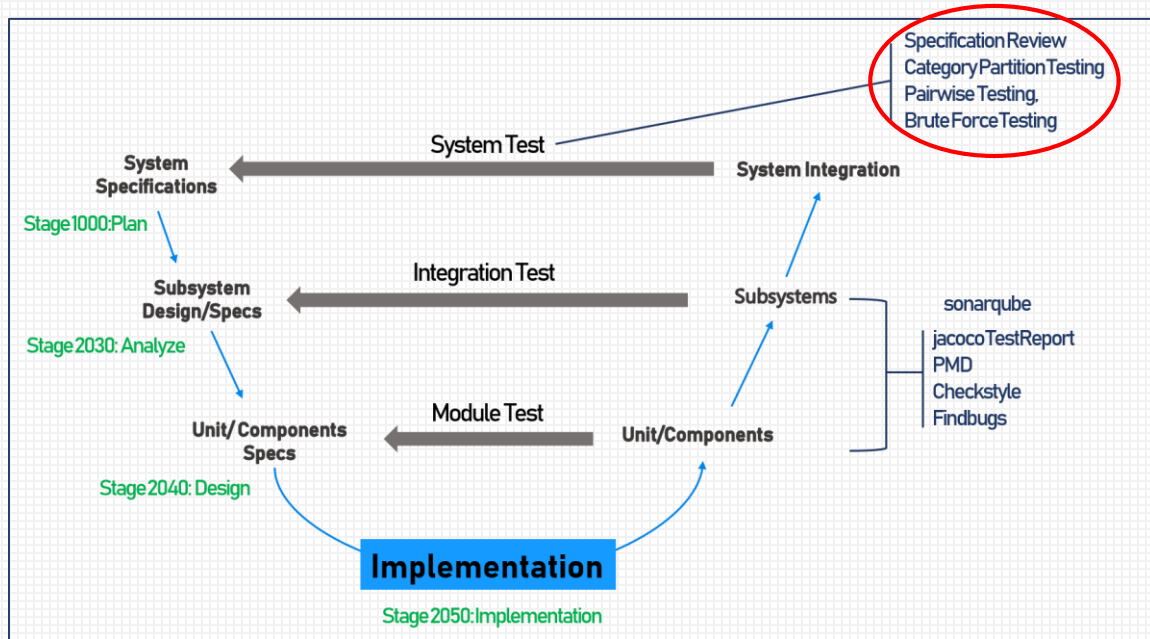
”

- 교수창

02. When Testing with SM Team ; Category Partitioning and Static Analysis?



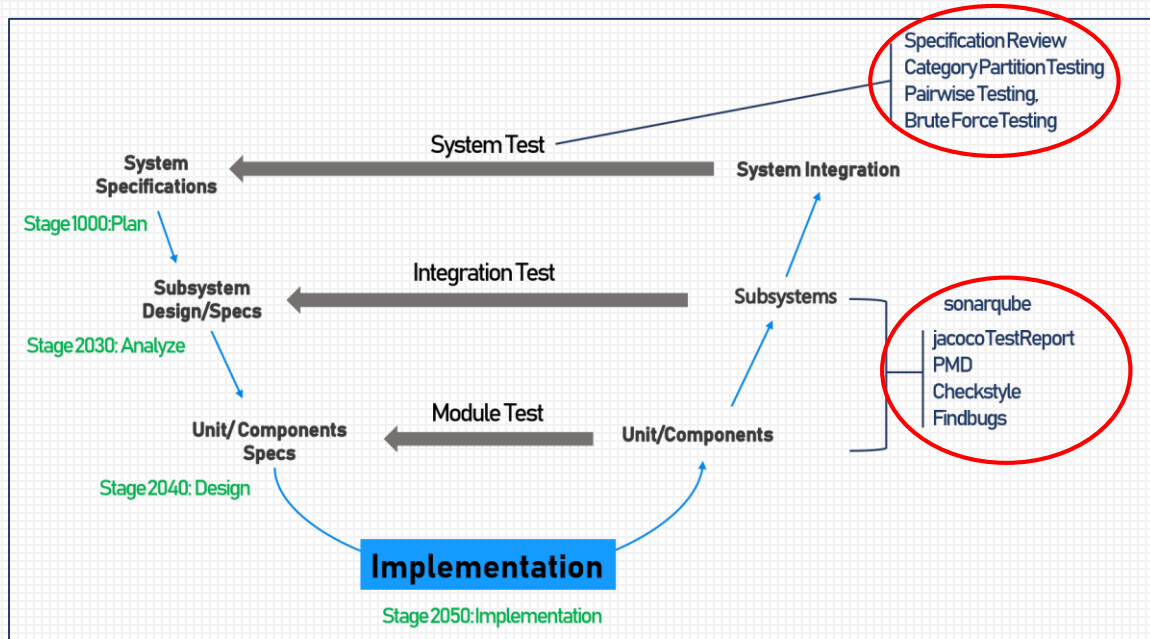
02. When Testing with SM Team ; Category Partitioning and Static Analysis?



CPT

- * Using Representative value ▷ Easy to catch Boundary Error.
- * TSL(tool)+Constrains ▷ Effective than work manually.
- * Practice several times is much helpful than read text.
- * Divide situation MECE will helpful.

02. When Testing with SM Team ; Category Partitioning and Static Analysis?



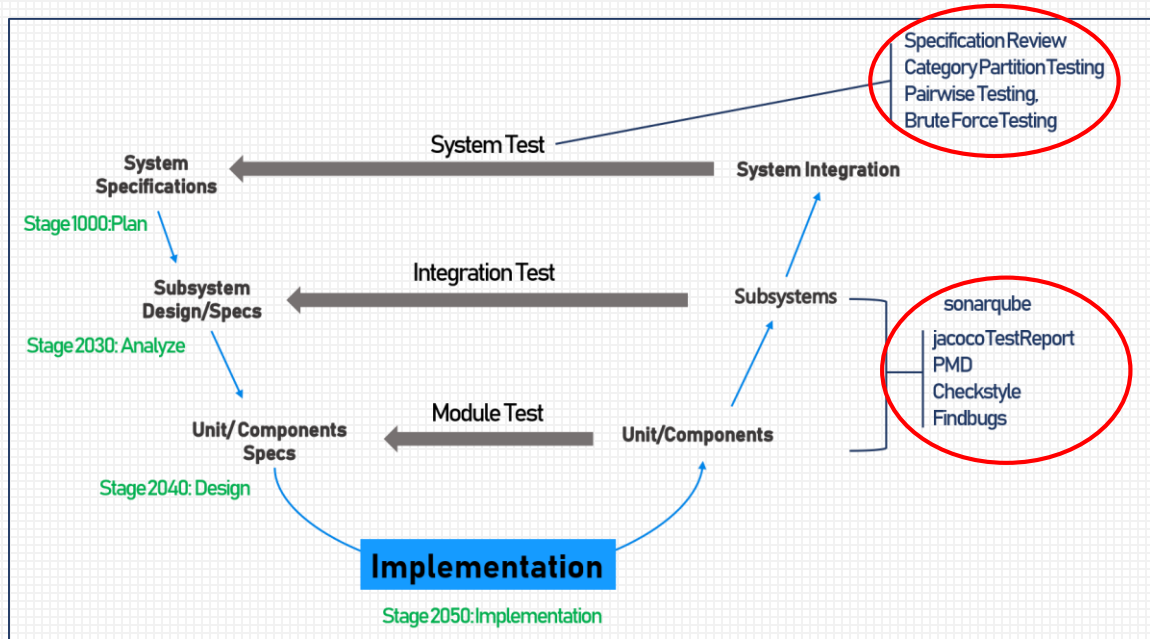
CPT

- * Using Representative value ▷ Easy to catch Boundary Error.
- * TSL(tool)+Constrains ▷ Effective than work manually.
- * Practice several times is much helpful than read text.
- * Divide situation MECE will helpful.

Static Analysis

- * Requirement coverage is more important.
- * Need to check convention for each project.
- * It gives us useful information. But, too much.
- * Anyway we have to review again manually.

02. When Testing with SM Team ; Category Partitioning and Static Analysis?



CPT

- * Using Representative value ▷ Easy to catch Boundary Error.
- * TSL(tool)+Constrains ▷ Effective than work manually.
- * Practice several times is much helpful than read text.
- * Divide situation MECE will helpful.

Static Analysis

- * Requirement coverage is more important.
- * Need to check convention for each project.
- * It gives us useful information. But, too much.
- * Anyway we have to review again manually.

Overall

We don't have plenty time to think about V&V model.
It is hard to implement theoretical aspects that we've learned.

02. When Testing with SM Team ; Thoughts about Team project _ Verification Team



It was so shame that we had a difficulty in communication although we both try hard.
If communication went smoothly and actively we could perform better
Also Wish there was a clear deadline when upload new version of documents,
and It would be nice if the reasons for the modified or unmodified parts.
Because we don't know there intention, been through tough days while doing second reviewing



I already took the software modeling class before, It was easy in some parts.
On the other hand, I could not point out all the mistakes because the amount was huge
Also, Wish there was a sharing session about the Planning.
If we knew the background of Why they choose to make that system
, we could communicate each other more effectively

02. When Testing with SM Team ; Thoughts about Team project _ Modeling Team



Through the project we can revising the documents, due to Verification team support.
And this experience make us more careful when writing a documents,



Because we don't have enough time, hard to communicate smoothly
and lead to discordance between code and documents.



After revising the documents we faced newly discovered mistakes.
So, we have to modify the code and documents again and again.
It makes hard to focus on developing process.

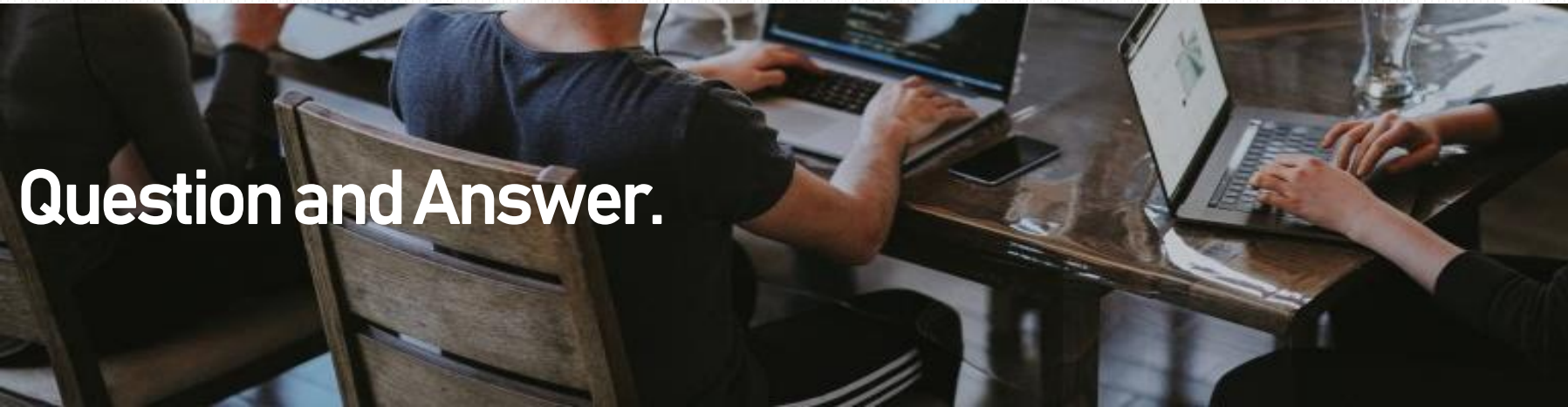


Part3. **Suggestion** for Future Lecture

01. Suggestions

; Software Verification & Software Modeling

- Teaching how to use tools or offering Example case would be better.
 - It is necessary for each team to start communication before developing.
 - Choosing tools are free, but it would be nice to have a basic default guide line.
 - Pointed out the theoretical part once more during the lab session would be helpful.
 - Since SM Team has not been taught about quality process, seems need to learn some details of the SV class.
 - Reducing the environment construction time, increasing the opportunity or time to test can help to get better skills.
-



Question and Answer.
