Visualized guideline model for measuring the Assessment Model based on an Simplified Test Maturity Model(TMM)

Jang, Woo Sung

SE Lab

selab.hongik.ac.kr

Advisor : R. Young Chul Kim

Outline

1. Motivation

2. Related works
   -Original TMM(Test Maturity Model)

3. Simplified Test Maturity Model (TMM)

4. Visualized Guideline Model of Simplified TMM

5. Conclusion

Motivation

- Why need the quality of Software?
- Why software is important?

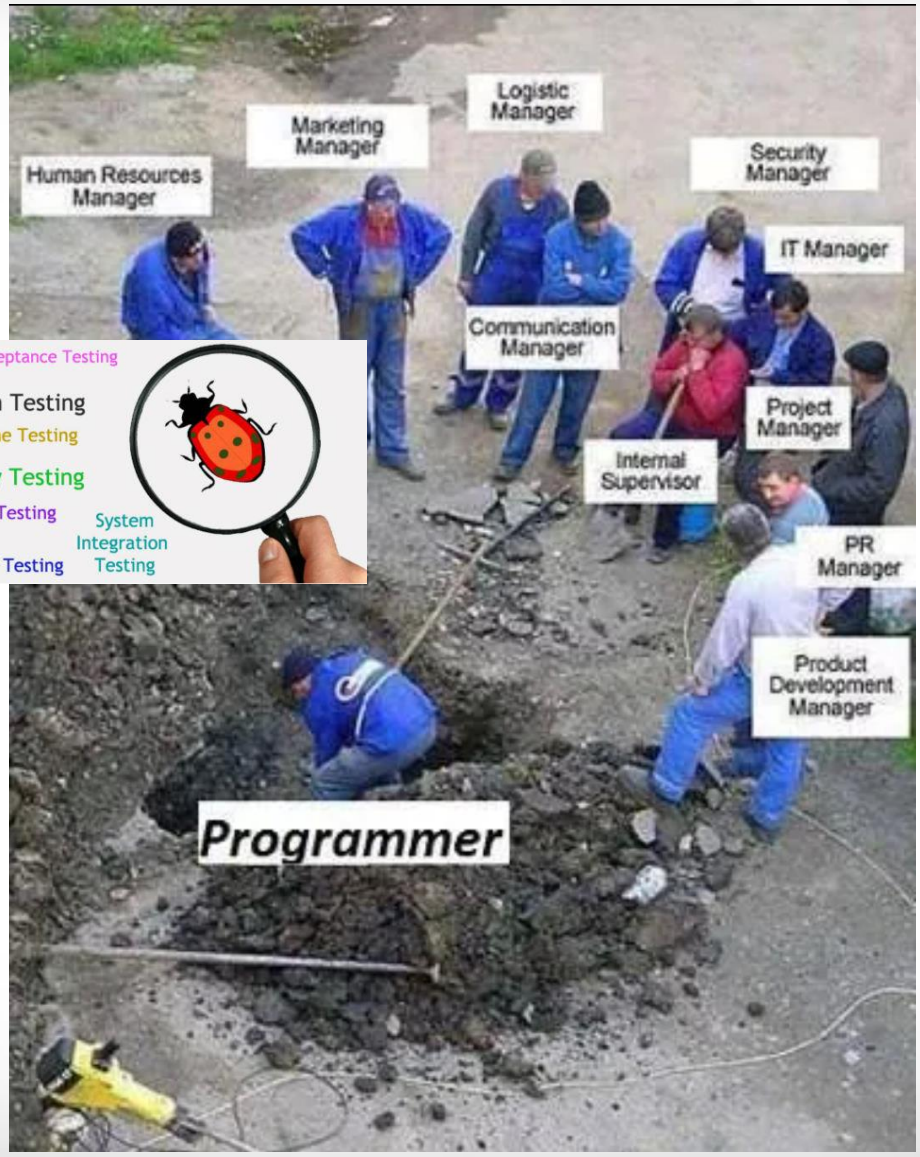
Accidents occurred by errors

	Accidents		Damage	Causes
Personal injury caused by SW Errors	2009 Washington Subway's Collision		• 9 person's death • Over 70 persons damage	• System error on automatic driving mode • Stop to work the break
	2014 America Southwest temporal suspension of landing service		• Delay Air 212 airplane • Cancel 21 airplanes	• LA air control center • On calculating U2 a reconnaissance plane on U2 Control system, stop system due to overload
Ignoring Warning for SW safety	2010 BP A fire on an oil ship		• British Petroleum (BP) • Leaking an oil and occurring a fire in an oil ship	• On needing to have space between Oil pipeline, • Give automatic alarm • BP to lose cost to delay construction, ignore warning
SW deficiency for Saftety	2015 young jong bridge car collision		• 106 accidents in collision • 2 person's death • Over 70 persons damage	• Over 100km speed, deficiency for safety

How to produce quality of Software?

- Who is important?

Too much work to developer



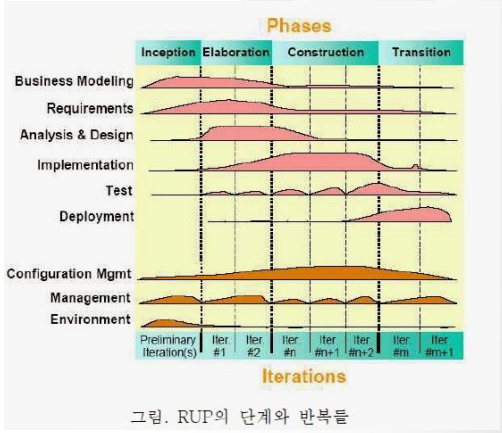


그림. RUP의 단계와 반복들

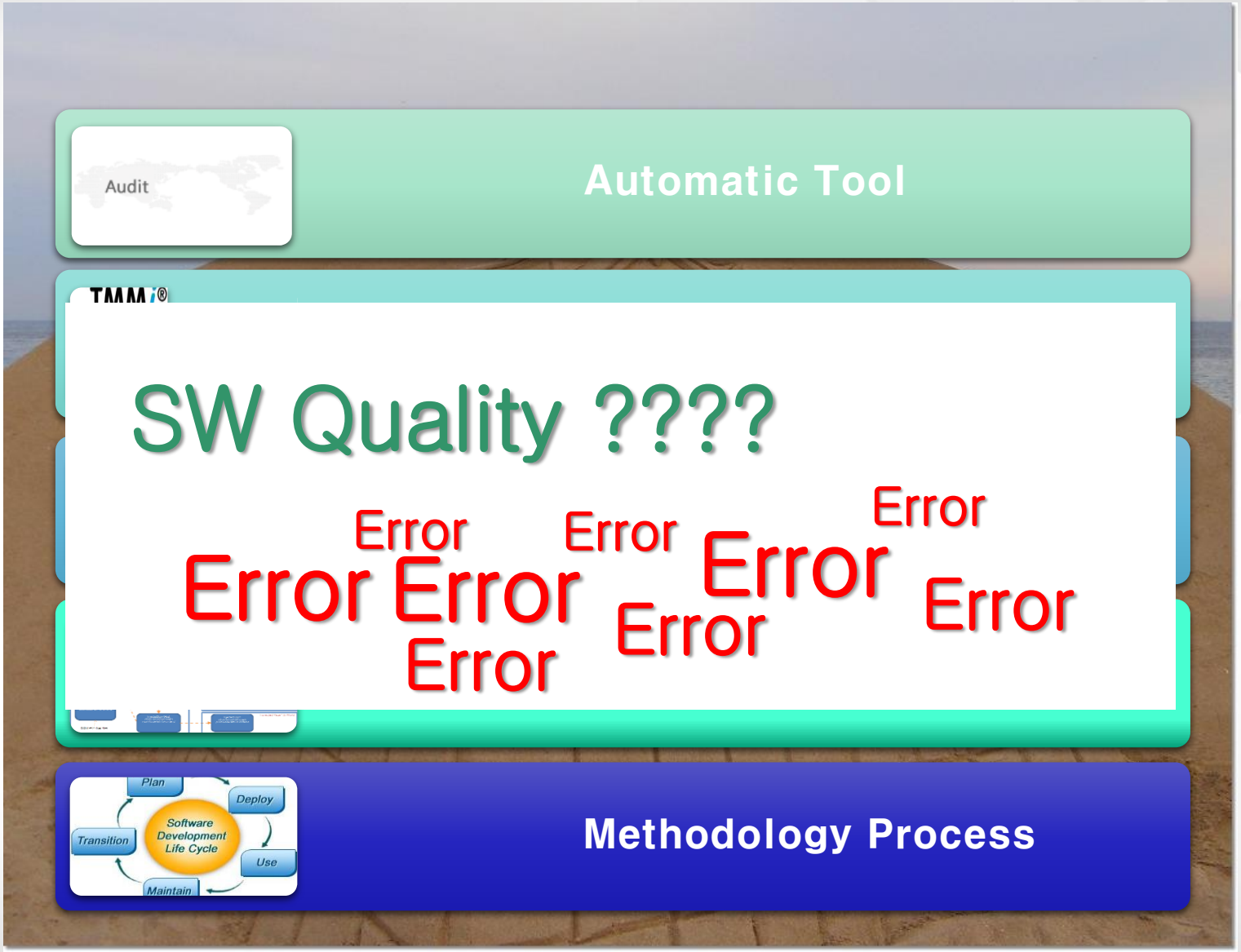
TMMi® FOUNDATION

Software Process

CMMI

GOOD Software

Need to work all ways for high quality SW

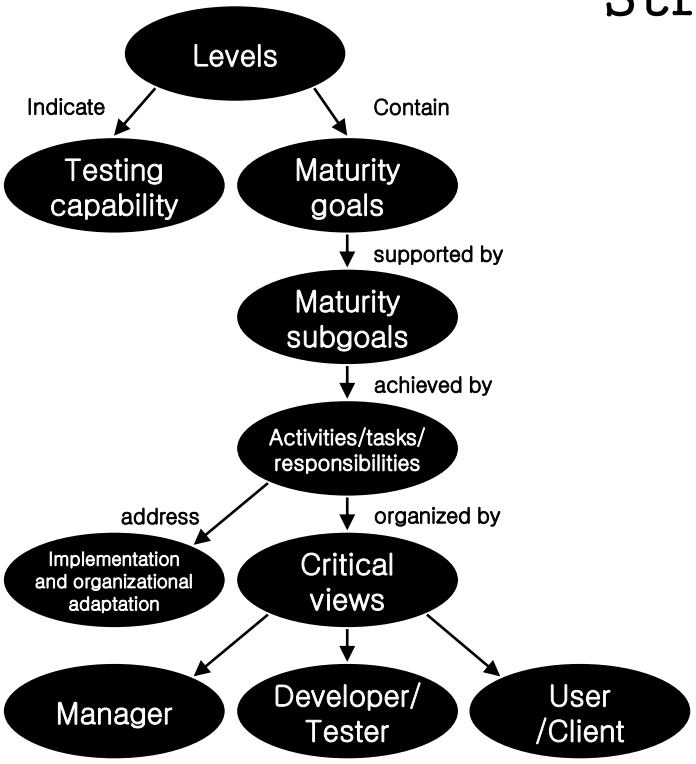


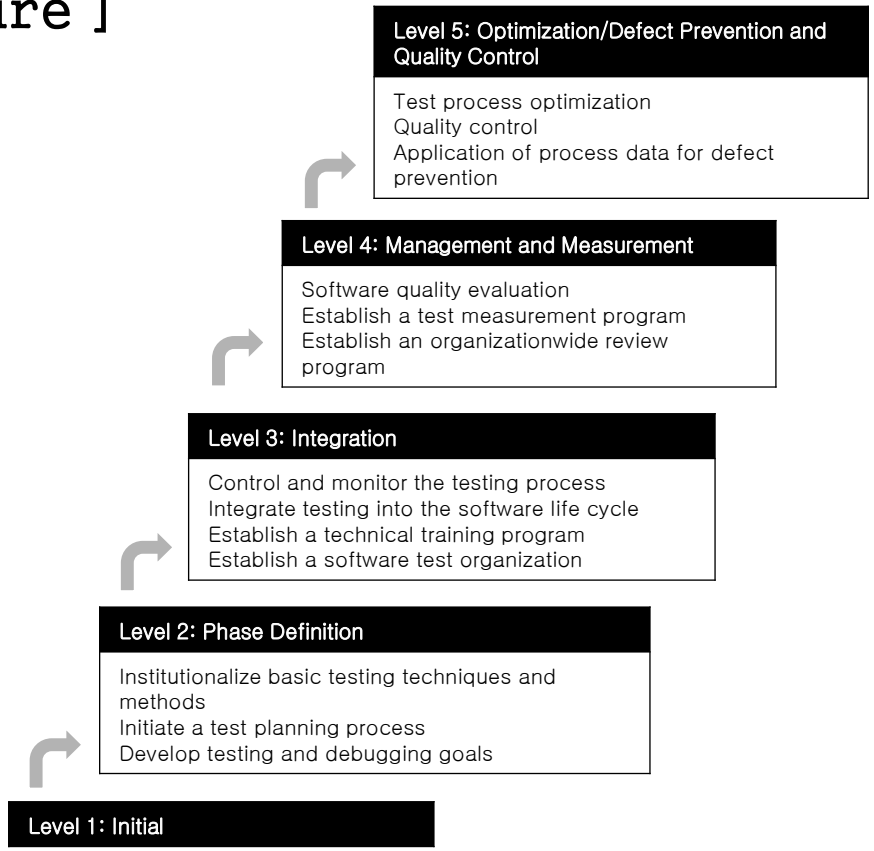
2. Related works(cont.)

- TMM(Test Maturity Model)
  - 1996 Dr.Burnstein at IIT, Chicago
  - based on CMM(Capability Maturity Model)
  - 5 level of maturity
  - Assessment Model
  - Process for Test Organization

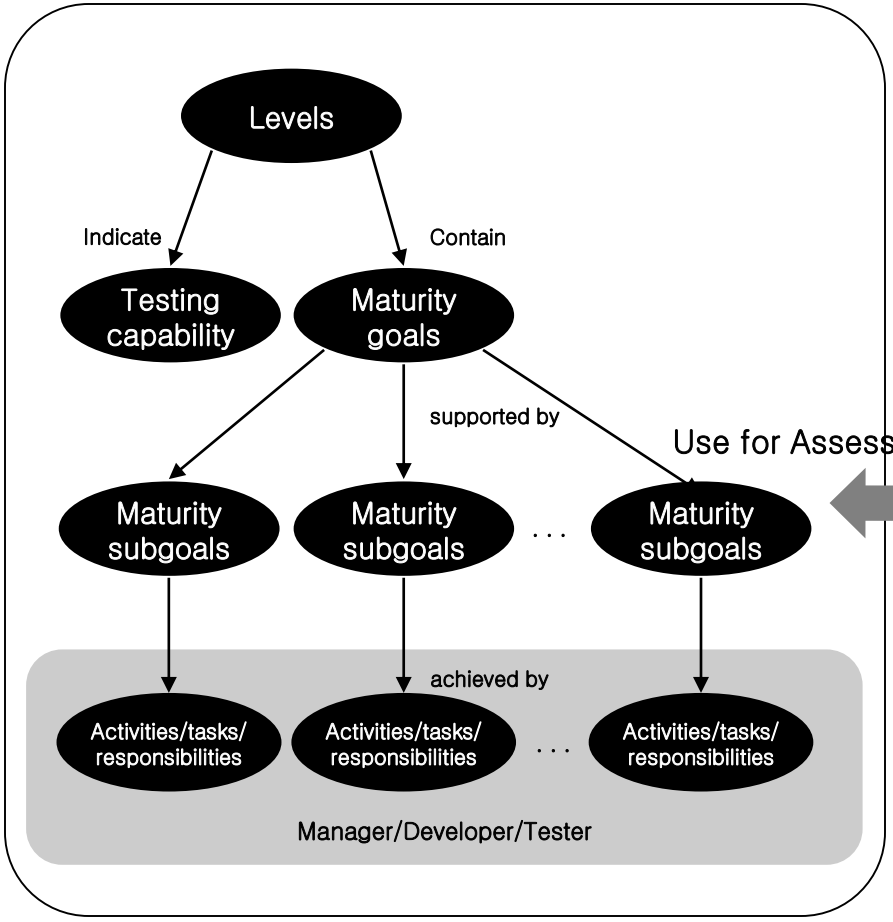
2. Related works(cont.)

[Original TMM Structure]



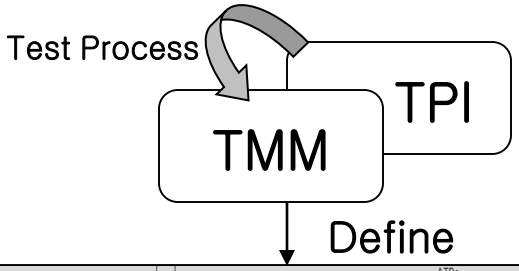


3. Simplified TMM



Simplified TMM Framework

Use for Assessment



Maturity SubGoal		ATRs					
			Managers		Developers		Users/Clients
2.1.1	An organizationwide committee(s) or group on testing and debugging is formed and provided with funding and support. The committee(s) develops, documents, distributes, and supports procedures, goals, and policies for testing debugging. The goals, policies, and procedures, once approved, are put in place and periodically reviewed.	1	Provide leadership, adequate resources, and funding to form the committee(team or task force) on testing and debugging. The committee makeup is managerial, with technical staff serving as comembers.	1	Work with management to develop testing and debugging policies and goals.	1	Users/clients give input/feedback on testing and debugging goals and policies when solicited by management. (These groups play an indirect role in the formation of an organization's testing/debugging goals and polices since these goals and policies reflect the organization's efforts to ensure customer/client/user satisfaction. Feedback from these groups should be encouraged by management and SQA. In general, the needs of its customers and the overall marketplace will have an impact on the nature of an organization's testing/debugging goals and policies.)
		2	Make available any pre-existing or sample testing/debugging policies and goals.	2	Participate in the team that oversees testing/debugging policy compliance and change management.		
		3	Assume a leadership role in testing/debugging policy development.	3	Develop test plans that are in compliance with testing policy.		
		4	Support the recommendations and policies of the committee by: —distributing testing/debugging goal/policy documents to project managers, developers/ testers, and other interested staff, and soliciting feedback from these groups; —appointing a permanent team to oversee compliance and policy change-making;	4	Become familiar with the approved set of testing/debugging goals and policies, keeping up-to-date with revisions and making suggestions for changes when appropriate.		
		5	Ensure necessary training, education, and tools to carry out defined testing/debugging goals and policies are made available.	5	Set testing goals for each project at each level of test that reflect organizational testing goals and policies.		
2.1.2	Testing and debugging policies/goals are reflected in project/test plans.	6	Promote the cultural changes needed to implement the testing/debugging policies.	6	Carry out testing activities that are in compliance with organizational policies.		
		7	Assign responsibilities for testing and debugging.	7	Work with project managers to develop a defect classification scheme and defect repository.		
2.1.3	A basic defect classification scheme is established, and a basic defect repository is put into place.	8	Encourage input/feedback from key users/client groups for testing/debugging policies.	8	Record defects occurring in each project in the defect repository.		
		9	Ensure that there is support for development of a simple defect classification scheme and a defect repository. The classification scheme and repository should be available for all project personnel to study and access.				
2.1.4	Simple testing and debugging measurements are identified and collected.	10	Ensure that developers (testers) are familiar with the defect classification scheme and record defects occurring for each project in the repository.	9	Participate in periodic reviews of testing/debugging policies and goals.		
		11	Periodically review testing and debugging goals and policies.				

Test Attributes to Maturity Level collation Matrix

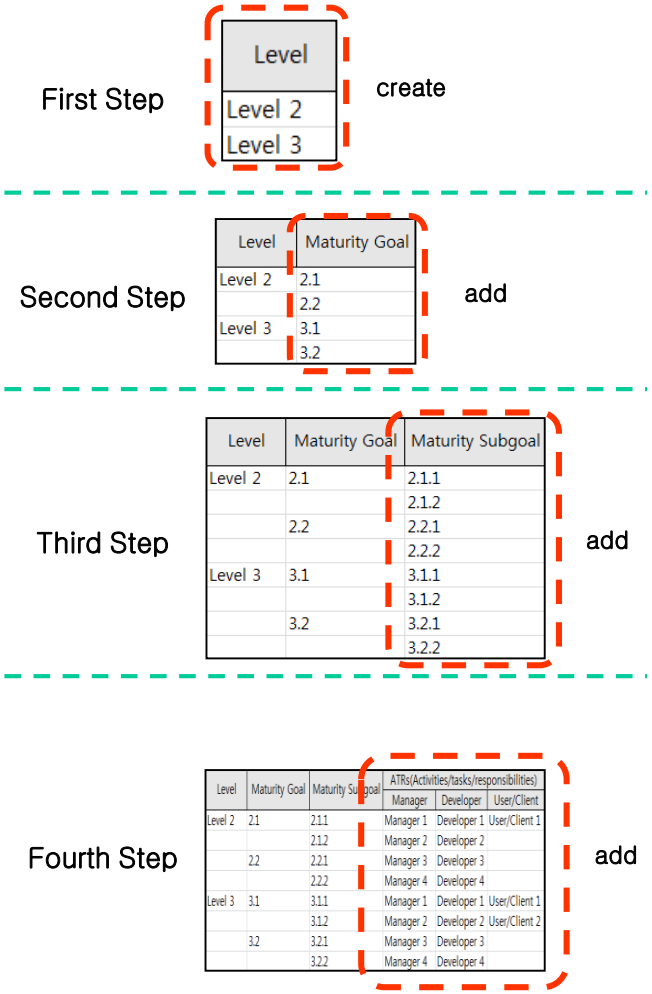
Enhanced TMM

How to make Simplified Test Maturity Model?

- Simplified TMM Development Process
  - Step 1
    - Changing the sentence of TMM documents into a table.
  - Step 2
    - Determining Goodness-of-fit of each TMM ATRs.
  - Step 3
    - Identifying relationship between the maturity subgoals and ATRs.
    - Merge each ATRs into the one ATRs.
  - Step 4
    - Adding TPI next attributes into TMM.

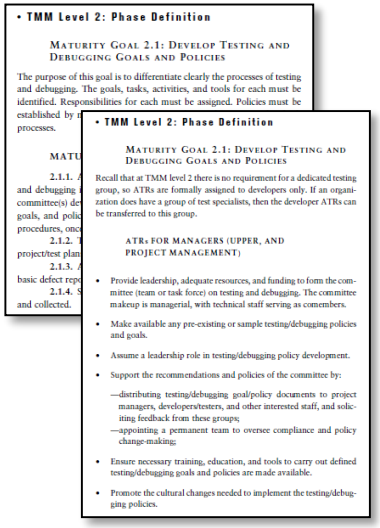
Simplified Test Maturity Model

- Step 1 – Changing the sentence of TMM document into a table
  - 1.1 Step
    - Changing the levels into a table.
  - 1.2 Step
    - Changing the maturity goals into a table.
  - 1.3 Step
    - Changing the maturity subgoals into a table.
  - 1.4 Step
    - Changing the ATRs(Manager, Developer, User) into a table.



Simplified Test Maturity Model

- Step 1 – make tablization with TMM documents of all Maturity Goals



• TMM Level 2: Phase Definition

MATURITY GOAL 2.1: DEVELOP TESTING AND DEBUGGING GOALS AND POLICIES

The purpose of this goal is to differentiate clearly the processes of testing and debugging. The goals, tasks, activities, and tools for each must be identified. Responsibilities for each must be assigned. Policies must be established by t… processes.

MATU…

2.1.1. A… and debugging … committee(s) de… goals, and polic… procedures, once…

2.1.2. T… project/test plan…

2.1.3. A… basic defect repo…

2.1.4. S… and collected.

• TMM Level 2: Phase Definition

MATURITY GOAL 2.1: DEVELOP TESTING AND DEBUGGING GOALS AND POLICIES

Recall that at TMM level 2 there is no requirement for a dedicated testing group, so ATRs are formally assigned to developers only. If an organization does have a group of test specialists, then the developer ATRs can be transferred to this group.

ATRs FOR MANAGERS (UPPER, AND PROJECT MANAGEMENT)

- Provide leadership, adequate resources, and funding to form the committee (team or task force) on testing and debugging. The committee makeup is managerial, with technical staff serving as comembers.
- Make available any pre-existing or sample testing/debugging policies and goals.
- Assume a leadership role in testing/debugging policy development.
- Support the recommendations and policies of the committee by:
  - —distributing testing/debugging goal/policy documents to project managers, developers/testers, and other interested staff, and soliciting feedback from these groups;
  - —appointing a permanent team to oversee compliance and policy change-making;
- Ensure necessary training, education, and tools to carry out defined testing/debugging goals and policies are made available.
- Promote the cultural changes needed to implement the testing/debugging policies.

TMM Document

Result

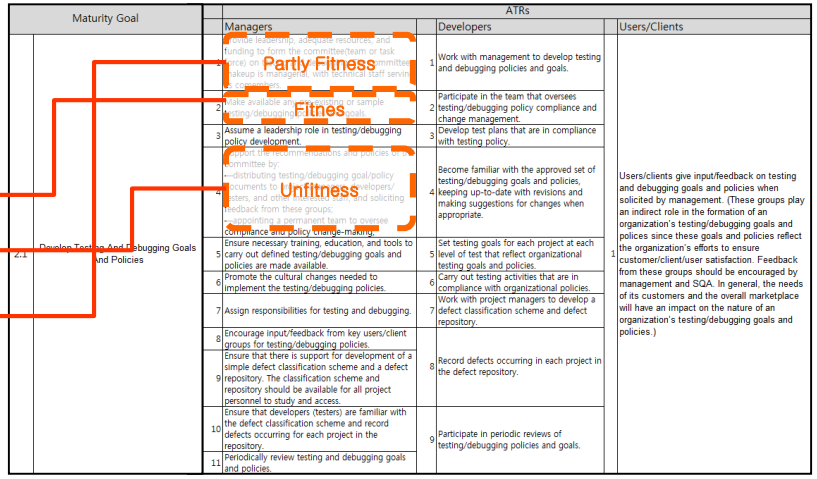
Level	Maturity Goal		Maturity SubGoal		ATRs								
					Managers			Developers			Users/Clients		
Level 2 Phase Definition	2.1	Develop Testing And Debugging Goals And Policies	2.1.1	An organizationwide committee(s) or group on testing and debugging is formed and provided with funding and support. The committee(s) develops, documents, distributes, and supports procedures, goals, and policies for testing debugging. The goals, policies, and procedures, once approved, are put in place and periodically reviewed.	1	Provide leadership, adequate resources, and funding to form the committee/team or task force) on testing and debugging. The committee makeup is managerial, with technical staff serving as comembers.	1	Work with management to develop testing and debugging policies and goals.	1	Users/clients give input/feedback on testing and debugging goals and policies when solicited by management. (These groups play an indirect role in the formation of an organization's testing/debugging goals and polices since these goals and policies reflect the organization's efforts to ensure customer/client/user satisfaction. Feedback from these groups should be encouraged by management and SQA. In general, the needs of its customers and the overall marketplace will have an impact on the nature of an organization's testing/debugging goals and policies.)
					2	Make available any pre-existing or sample testing/debugging policies and goals.	2	Participate in the team that oversees testing/debugging policy compliance and change management.				
					3	Assume a leadership role in testing/debugging policy development.	3	Develop test plans that are in compliance with testing policy.				
					4	Support the recommendations and policies of the committee by: —distributing testing/debugging goal/policy documents to project managers, developers/testers, and other interested staff, and soliciting feedback from these groups; —appointing a permanent team to oversee compliance and policy change-making;	4	Become familiar with the approved set of testing/debugging goals and policies, keeping up-to-date with revisions and making suggestions for changes when appropriate.				
					5	Ensure necessary training, education, and tools to carry out defined testing/debugging goals and policies are made available.	5	Set testing goals for each project at each level of test that reflect organizational testing goals and policies.				
			2.1.2	Testing and debugging policies/goals are reflected in project/test plans.	6	Promote the cultural changes needed to implement the testing/debugging policies.	6	Carry out testing activities that are in compliance with organizational policies.				
					7	Assign responsibilities for testing and debugging.	7	Work with project managers to develop a defect classification scheme and defect repository.				
			2.1.3	A basic defect classification scheme is established, and a basic defect repository is put into place.	8	Encourage input/feedback from key users/client groups for testing/debugging policies.	8	Record defects occurring in each project in the defect repository.				
					9	Ensure that there is support for development of a simple defect classification scheme and a defect repository. The classification scheme and repository should be available for all project personnel to study and access.						
			2.1.4	Simple testing and debugging measurements are identified and collected.	10	Ensure that developers (testers) are familiar with the defect classification scheme and record defects occurring for each project in the repository.	9	Participate in periodic reviews of testing/debugging policies and goals.				
					11	Periodically review testing and debugging goals and policies.						

S_TMM Tablization

Simplified Test Maturity Model

- Step 2 – Determining Goodness-of-fit of each TMM ATRs (Activities/Tasks/Responsibilities)
  - Decide suitability of TMM ATRs for Korean Small & Medium Company
  - Describe decision method as follows.

decision	description	
Fitness	available attribute	Accept
Unfitness	unavailable attribute	Unaccept
Partly fitness	partly available attribute	Accept



Simplified Test Maturity Model

❑ Step 2 – Goodness-of-fit decision Result of TMM ATRs

[Original TMM]

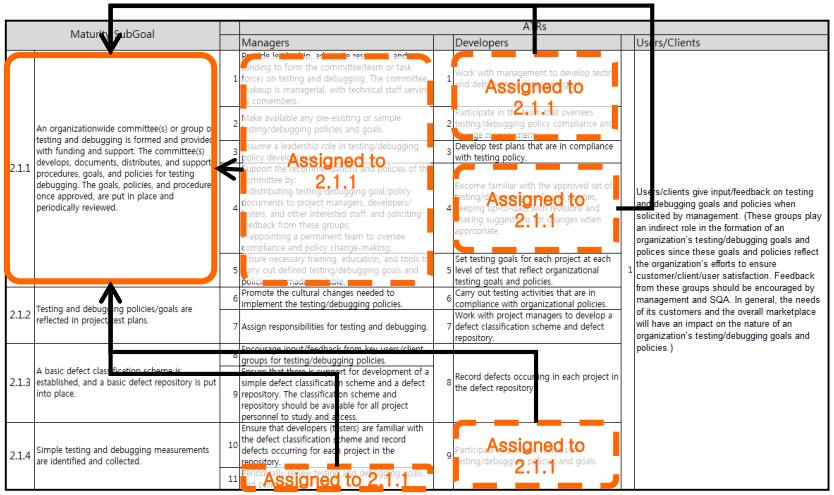
Level	Level 2			Level 3			
Maturity Goal	2.1	2.2	2.3	3.1	3.2	3.3	3.4
Manager's ATRs	11	19	16	17	10	11	12
Test Manager's ATRs	–	–	–	–	–	–	13
Developer/Tester 's ATRs	9	10	17	21	6	10	12
Total	20	29	33	38	16	21	37

[Simplified TMM]

Level	Level 2			Level 3			
Maturity Goal	2.1	2.2	2.3	3.1	3.2	3.3	3.4
Manager's ATRs	9	17	12	14	7	9	9
Test Manager's ATRs	–	–	–	–	–	–	13
Developer/Tester's ATRs	8	9	17	20	5	10	12
Total	17	26	29	34	12	19	25

Simplified Test Maturity Model

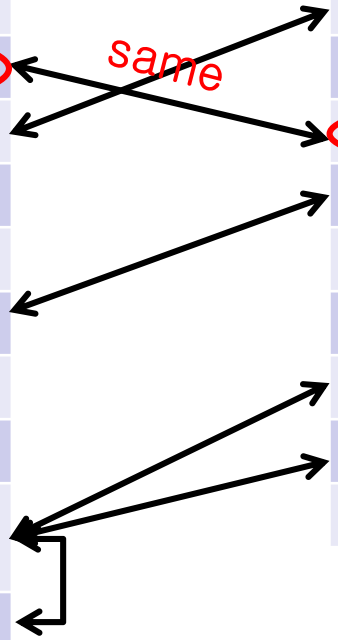
- Step 3 – Identifying relationship between the maturity subgoals and ATRs
  - ATRs of TMM are connected with maturity goals.
  - But ATRs of Simplified TMM are connected with maturity subgoals.
  - If all the ATRs be achieved, maturity subgoal be achieved.
  - If all the maturity subgoals be achieved, maturity goal be achieved.



Simplified Test Maturity Model

- Step 3 – Merge each ATRs into the one ATRs
  - In ATRs of maturity subgoal, merge if similar activities on each Critical View.
  - In "unfitness" case, delete attributes.

Maturity Subgoal	ATRs (Managers)	
2.1.1	1	Unfitness
	2	Integrate Developer 4
	3	Integrate Developer 1
	4	Unfitness
	5	Fitness
	11	Integrate Developer 9
2.1.2	7	Fitness
	8	Unfitness
2.1.3	9	Integrate Manager10, Developer 7, 8
	10	Integrate Manager 9
2.1.4	6	Unfitness



Maturity Subgoal	ATRs (Developers)	
2.1.1	1	Integrate Manager 3
	2	Move Level 3
	4	Integrate Manager 2
	9	Integrate Manager 11
2.1.2	3	Fitness
	5	Fitness
2.1.3	7	Integrate Manager 9
	8	Integrate Manager 9
2.1.4	6	Fitness

Simplified Test Maturity Model

- Step 4 – Adding TPI next attributes into TMM
  - Compare TMM with TPI Next.
  - Identify *each deficient test activity* in maturity subgoal of TMM.
  - Add *TPI Next attribute* into the deficient test activity of TMM.

* TPI: Test Process Improvement

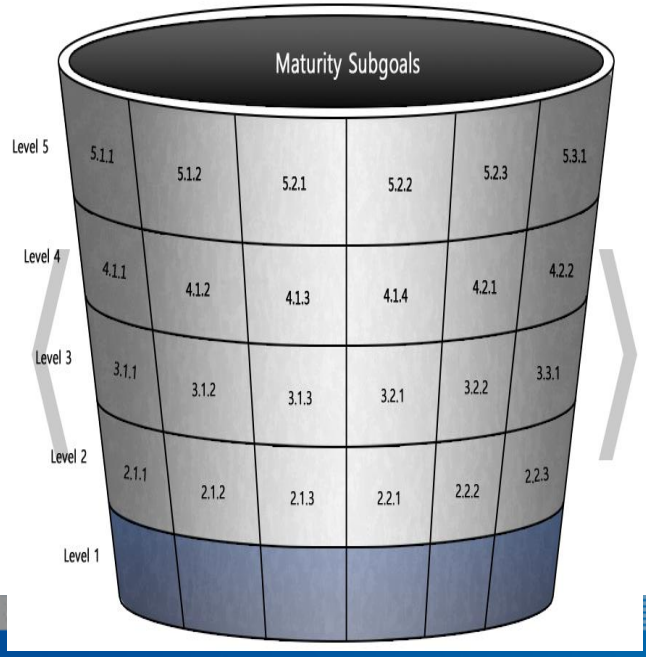
Simplified Test Maturity Model

- We simplified with TMM.
  - The comparison result were as follows.

	TMM		Simplified TMM	
	Level 2	Level 3	Level 2	Level 3
Maturity Goal	3	4	3	4
Maturity Subgoal	13	12	14	12
ATR (Activities/tasks/ responsibilities)	91	118	75	102

Visualized guideline model for measuring an assessment model base on Simplified TMM

- Easy to assess test organization with assessment model
- Identify what to need and what to do more activity
- Then help and guide to enhanced level of TMM for small and medium companies in Korea
- We just check each question in the assessment model at right menu.



Maturity Subgoal	2.1.1
Procedures, objectives, policy development documents	○ Yes ◉ No
Important equipment list	○ Yes ◉ No
Resource management plan	○ Yes ◉ No

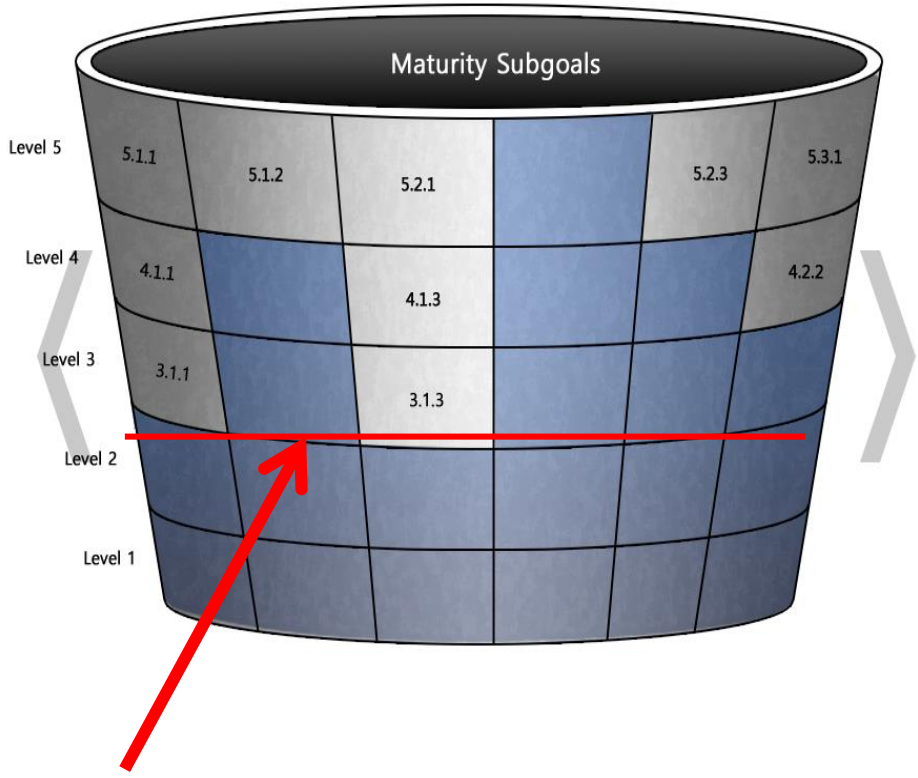
Maturity Subgoal	2.1.2
Organized workforce planning	○ Yes ◉ No
Test scenario	○ Yes ◉ No
Test plan	○ Yes ◉ No
Based on test passed	○ Yes ◉ No

Maturity Subgoal	2.1.3
Defect life cycle document	○ Yes ◉ No
Defect classification scheme documents	○ Yes ◉ No
Use of defective store	○ Yes ◉ No

Save Cancel

Visualized guideline model

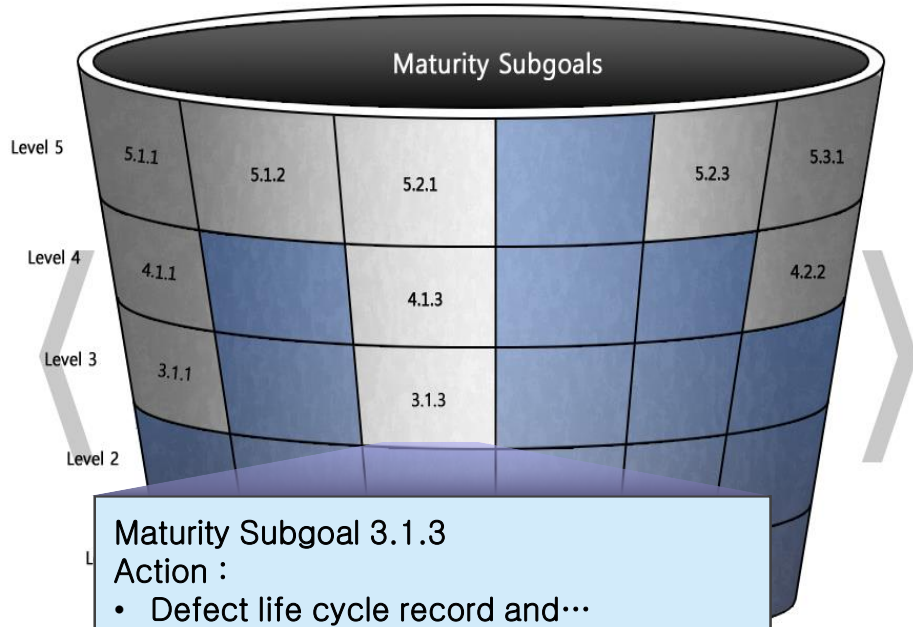
- Like Vitamin Bucket Model



Level	Maturity Subgoal												
Level 5	5.1.1	5.1.2	5.2.1		5.2.3	5.3.1	5.3.2	-	-	-	-	-	-
Level 4	4.1.1		4.1.3			4.2.2	4.2.3	4.3.1	4.3.2	4.3.3	4.3.4	-	-
Level 3	3.1.1		3.1.3				3.3.2	3.3.3			3.4.3	3.4.4	-
Level 2													
Level 1													

Level 2

Easily guiding how to enhance level up



Maturity Subgoal 3.1.3
Action :
- Defect life cycle record and…
- Development defect classification…
- Fault recording and documentation

Output:
- Defect life cycle document
- Defect classification scheme documents
- Use of defective store

Level	Maturity Subgoal												
Level 5	5.1.1	5.1.2	5.2.1		5.2.3	5.3.1	5.3.2	-	-	-	-	-	-
Level 4	4.1.1		4.1.3			4.2.2	4.2.3	4.3.1	4.3.2	4.3.3	4.3.4	-	-
Level 3	3.1.1		3.1.3				3.3.2	3.3.3			3.4.3	3.4.4	-
Level 2													
Level 1													

Maturity Subgoal	3.1.3
Action	• Defect life cycle record and definition • Development defect classification system and fault repository • Fault recording and documentation
Output	v Defect life cycle document ▪ Required items: faulty identifier, testers, test date, test subject, title, likelihood, severity, fault details, fault handling results, handler, processing date, type of processing, process descriptions, process validation results, verifier, verification date, test results, validate description X Defect classification scheme documents ▪ Required items: used to define the defect classification system to suit the company's test environment. v Use of defective store ▪ Establish a defect related to data storage and staff to establish a plan to take advantage of using faulty data.

Conclusion

We propose Simplified TMM

- Help to measure TMM for test organization
- Guide what need to work more maturity level
- With our model, will apply two IT companies with TTA in 2016

Thank you