

# SQALE - Remediation and Non-Remediation Function

by  
Ramraj S ME.,  
Assistant Professor  
Department of Software Engineering

10 -FEB -2016

# Agenda

- ▶ Recap of Software quality measurement
- ▶ SQALE
- ▶ Remediation Function
- ▶ Non-Remediation Function

# SQALE - Quality Framework or Model

## Principles

- ▶ quality of the source code is a non-functional requirement
- ▶ Formalising requirements in relation to the quality of the source code
- ▶ Assessing the quality of a source code
- ▶ Cost remediation to meet the Quality
- ▶ assesses the importance of a non-conformity quality
- ▶ SQALE Methods Quality Model is orthogonal
- ▶ SQALE Method uses addition for aggregating the remediation costs, the non-remediation costs and for calculating its indicators

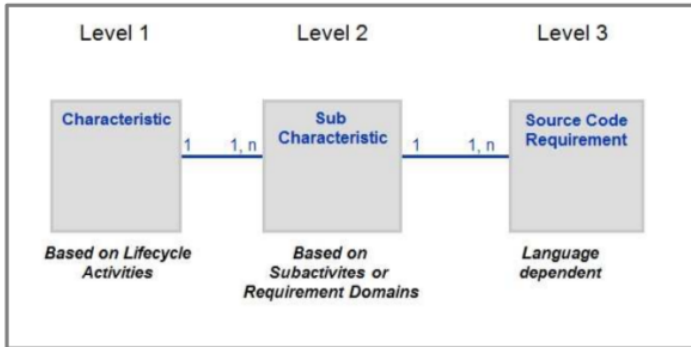


Figure : Classification



Figure : First Level

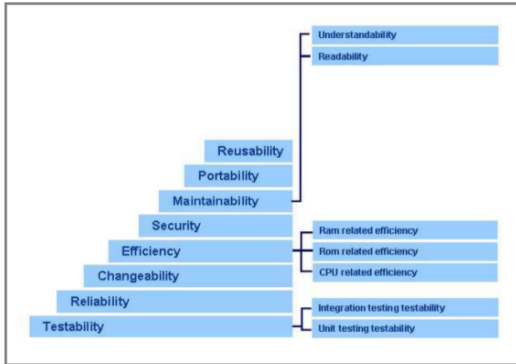


Figure : Second Level

Characteristic	Subcharacteristic
Reusability	Extractability
Reusability	Conciseness
Reusability	Stability
Portability	Language related portability
Portability	Time zone related portability
Portability	Hardware related portability
Portability	External application related portability
Portability	Compiler related portability
Portability	OS related portability
Maintainability	Understandability
Maintainability	Readability
Security	OS related security
Security	User related security
Security	Statement related security
Efficiency	ROM related efficiency
Efficiency	RAM related efficiency
Efficiency	CPU related efficiency
Changeability	Architecture related changeability
Changeability	Logic related changeability
Changeability	Data related changeability
Reliability	Fault tolerance
Reliability	Architecture related reliability
Reliability	Resource related reliability
Reliability	Synchronization related reliability
Reliability	Statement related reliability
Reliability	Logic related reliability
Reliability	Data related reliability
Testability	Integration Testing testability

Figure : Example

Characteristic	SubCharacteristic	Generic Requirement Description
Maintainability	Understandability	No unstructured statements (goto, break outside a switch...)
Maintainability	Understandability	No use of "continue" statement within a loop
Maintainability	Understandability	File comment ratio (COMR) > 35%
Maintainability	Readability	Variable name start with a lower case letter
Maintainability	Readability	The closing brace '}' is on a standalone line
Maintainability	Readability	The code follow constant indentation rules
Maintainability	Readability	File size (LOC) <1000
Maintainability	Readability	No commented-out code
Efficiency	RAM related efficiency	Class depth of inheritance (DIT) <8
Efficiency	RAM related efficiency	No unused variable, parameter or constant in code
Changeability	Architecture related changeability	Class weighted complexity (WMC) <100
Changeability	Architecture related changeability	Class specification does not contains public data
Changeability	Logic related changeability	If, else, for, while structures are bound by scope
Changeability	Data related changeability	No explicit constants directly used in the code (except 0,1, True and False)
Reliability	Fault Tolerance	Switch' statement have a 'default' condition
Reliability	Logic related reliability	No assignment '=' within 'if' statement
Reliability	Logic related reliability	No assignment '=' within 'while' statement
Reliability	Logic related reliability	Invariant Iteration index
Reliability	Data related reliability	No use of uninitialized variables
Testability	Integration level testability	No "Swiss Army Knife" class antipattern
Testability	Integration level testability	Coupling between objects (CBO) <7
Testability	Unit Testing testability	No duplicate part over 100 token
Testability	Unit Testing testability	Number of ind. test paths within a module (v(G)) <11
Testability	Unit Testing testability	Number of parameters in a module call (NOP) <6

Figure : For JAVA



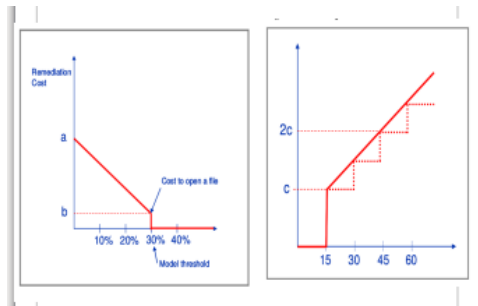
# Remediation Function

- ▶ SQALE model is supported by various analysis tool which will give different measures of the given unit of code.
- ▶ Remediation function objective is to normalise the findings to meet the quality requirement.
- ▶ The function consider the remediation cost for making the code to meet that.
- ▶ Remediation function uses a multiplicative factor that corresponds to the average remediation cost unit for bringing the code into conformity
- ▶ The value of this factor will be high or low depends on the change.
- ▶ The set of remediation functions associated to a Quality Model constitutes a Technical Debt estimation model

# Example - Remediation Function

NC Type Name	Description	Sample	Remediation Factor
Type1	Corrigible with an automated tool, no risk	Change in the indentation	0.01
Type2	Manual remediation, but no impact on compilation	Add some comments	0.1
Type3	Local impact, need only unit testing	Replace an instruction by another	1
Type4	Medium impact, need integration testing	Cut a big function in two	5
Type5	Large impact, need a complete validation	Change within the architecture	20

Figure : Remediation factor Calculation



# Non-Remediation Function

- ▶ Non-Remediation function calculates about the non-remediation cost.
- ▶ Non-Remediation cost relates to non-conformity.
- ▶ It calculates the non-remediation cost and the impact of the non-conformity in the life cycle.
- ▶ It also analysis the damages of the non-conformity of the quality requirement in the later stages of the life cycle

# Example - Non-Remediation Function

NC Type	Description	Sample	Non-Remediation Factor
Blocking	Will or may result in a bug	Division by zero	5 000
High	Will have a high/direct impact on the maintainance cost	Copy and paste	250
Medium	Will have a medium/potential impact on the maintainance cost	Complex logic	50
Low	Will have a low impact on the maintainance cost	Naming convention	15
Report	Very low impact, it is just a remediation cost report	Presentation issue	2

Figure : Non-Remediation

# SQALE - OVERVIEW

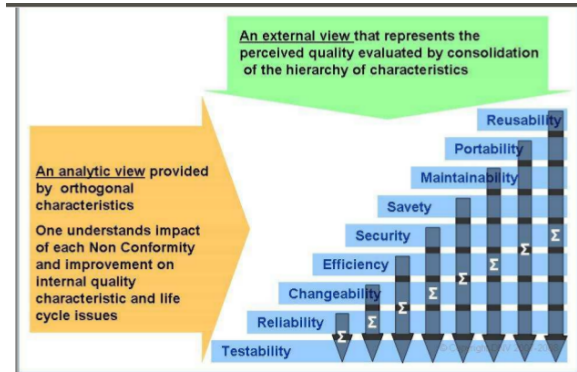


Figure : SQALE overview

# References I

- [1] Jean-Louis Letouzey, "*The SQALE Method Definition Document*", Chapter 17

Thank you