Software Metrics Proposal for Conformity Checking of Class Diagram to SOLID Design Principles

Intan Oktafiani and Bayu Hendradjaya

September 18, 2018

Software Metrics Proposal for Conformity Checking of Class Diagram to SOLID Design Principles

***Abstract*— Software design quality are the most important thing at this time to compete with the rapid development of technology. One of quality factor is the ability of software to be maintained. To produce quality, measurement at the design stage is the way to achieve it. The SOLID Design Principles is one of the object-oriented design guidelines to meet software quality factors to be more easily understood, more flexible, easy to maintain, and testable. The method used in this study is to analyze the relationship between the concepts of SOLID principles with class diagram metrics, the metrics for each principle are produced and the measurement techniques for class diagrams. Proposed metrics for SRP is VSRP (Value of SRP) as well as other principles: OCP - VOCP (Value of OCP), LSP - VLSP (Value of LSP), ISP - VISP (Value of ISP) and DIP - VDIP (Value of DIP). Metric validation is done by using 15 class diagrams from 7 software applications. The metric result is searched for correlation with the results by experts using Spearman's Rank Correlation. From the validation, it was found that 4 metrics have a strong, significant and unidirectional correlation. SRP's Metrics is a weak and insignificant correlate metric. We also found from the validation that there are correlation between metrics, that are between SRP Metrics with LSP Metrics, and LSP with ISP Metrics. The correlation shows that when the value of one metrics increases, the other metrics also rise. In this paper we develop Metrics to measure the conformity of class diagram to SOLID principles. The aim of this work is to help developers so that they can be re-checked and prevented as early as possible so as to minimize errors at the next stage of development.**

Keywords—SOLID Principles, object-oriented, Metrics, Class Diagrams.

I. Introduction

Design is the place where quality is fostered in software engineering. Design provides us with representations of software that can be assessed for quality. Software design serves as the foundation for all the software engineering and software support steps that follow. Without design, we risk building an unstable system that will fail when small changes are made; one that may be difficult to test; one whose quality cannot be assessed until late in the software process, when time is short and many resources have already been spent [1]. In addition, the application of quality at the design stage can minimize maintenance costs. The cost of repairing defects increases exponentially as the software develops through the development life cycle. The cost of fixing defects after release is very significant: up to 30 times more than if you catch it in the design and architecture phase [2].

The SOLID Principles design allows the management of most software design quality problems. This set of principles can provide an understanding of design to avoid bad design symptoms or what is known as "Bad Design", helping to reduce code complexity, improve readability, extendability, maintainability, reduce errors, implement reusability, achieve better testability, and reduce tight coupling in object-oriented software [3]. Based on observations by Johannes et al. (2018) of 104 engineers and architects who were concerned about these principles, they were informed that there were difficulties in following this principle in software design. One reason is the description of design principles is still too vague to be understood and applied correctly [4].

Class diagram of Unified Modeling Language (UML) as statistical diagram, design a system by showing classes, attributes, methods and relationships between objects. Object Oriented concept such as abstraction, encapsulation, inheritance, and polymorphism mean much for this research. Therefore, measures and metrics class, class hierarchy, and class relationshios will be invaluable to a software engineer who must assess design quality. Each of the characteristics can be used as the basis for measurement [1]. There are many OO Metrics that have been found such as CK Metrics, Li and Henry's Metrics, Lorenz and Kidd's Metrics, Bansiya et al. Metrics [5].

This paper address the relation of SOLID Design Principles and class diagram's metrics. We proposed SOLID metrics which can be one or more combined metrics with various calculation based on the analysis of the theory description of each principle and technic to get the conformity checking of the class diagram to the principles.

II. Related Works

A. SOLID Principles

SOLID was introduced by Robert C. Martin. This principle was chosen because these are principle for the management of object-based software. If you do not follow the SOLID Principle [3], the software will have a tendency to tight code coupling with other modules. Rigorous coupling produces code that tends to be difficult to test, code duplication, new bugs when fixing other bugs, and many unknown problem in the application development cycle.

The following are 5 SOLID design principles:

1) *SRP – The Single Responsibilities Principles*

"*A class should have only one reason to change*."

This principle is based on the work of Tom DeMarco and Meilir Page-Jones, regarding cohesion. Bad design is when a class has functions that are not cohesive or unrelated, it shows that too much responsibility is assigned to a class, so class design will show low cohesion.

2) *OCP – The Open Close Principles*

"*A module should be open for extension but closed for modification*."

One of the criteria that can be reviewed to detect design compliance with the OCP principle is the possibility of a conditioning in a method that involves a behavior with a type function but with implementation that varies. When that behavior is found, made it into an abstract method in the baseclass [4]. So it can prevented from changing to the old code when there is the addition of new components, but only extends the abstract method.

3) *LSP – The Liskov Subtitution Princples*

"*Subclasses should be substitutable for their base classes*."

This principle manages the hierarchy of inheritance to keep it in line with the OCP. Substitutability is if S Subclass is from T, then the type T object can be replaced with an object type S. The baseclass contract must be understood properly and clearly. In other words the derived class must really substitute its base class.

According to John Dooley (2011) [9], things that can be detected that a design violates LSP if:

a. The subclass doesn't implement the abstract behavior
b. Subclass modifies
c. The subclass creating an exception
d. The subclass implements an empty class

4) ISP – The Interface Segregation Principles

"*Many client specific interfaces are better than one general purpose interface*"

Interfaces must be separated according to their respective context. Interfaces must be specific, many interfaces that each manage one thing better than one interface that seems "fat", it is to avoid ineffectiveness of resources in the system that will be created. The implemen- tation class must not be forced to depend on interfaces that have functions that are not needed.

5) DIP – The Dependency Inversion Principles

"*Depend upon Abstractions. Do not depend upon concretions*"

High-level modules should not depend on low-level modules. For simplicity we can state that when designing interactions between high and low level modules, interaction must go through abstracts between them.

B. Related Works

Many related works support this research such as Class Cohesion Metric namely CAMC (Cohesion Among Methods in Class Metric*)* [6] which is very related to one of the SOLID principles namely Single Responsibility Principle which will be discussed in the next point.

One of similar research is paper that produce Design Quality Model (DQM) [7] that successed to measure Design Principles based on code compliance with implementation of best practice designs. This Principles is a survey results on potential candidates in their fields against the level of importance of the Design Principles. Measurements in this study were carried out at the implementation stage. Where maintenance costs are 5x greater than if errors are identified at the design stage.

The others studies have more common things. Where measurements are made at the design stage and the principles focus is SOLID Principles, but the basis of measurement used is algebraic operations with various mathematical predicates [8].

III. PROPOSED METRICS

Based on the literature, we analyst how to measure the value of the design conformity of class diagram design, against the SOLID Principles. There are several things and assumptions regarding class diagram measurements:

a. The class diagram that can be measured is a detailed class diagram, where the component diagram has a parameterized method and relations between classes.

b. Abstract methods that have not been implemented are required to be written again with italic font types in the subclass, subclass change into abstract class. If it is not written, then it's contain return false or exception.

c. The concrete method of baseclass abstract does not need to be rewritten in the subclass, if it is done it is considered as an action to override method.

d. Defining abstract methods must have the same component (name, parameter, and return type) as the baseclass.

e. Play class is not included in the measurement.

So the following are the results of an analysis, all of value on range 0-1 when 1 is a perfect value:

A. *Metrics of Single Responsibility Principles (SRP)*

This principle is about the cohesion of a class, the class component must be interrelated. Then we can measure the SRP value of a class by calculating the Metric value CAMC (Cohesion Among Methods of Class)[14]. This calculation can be done when all prototype methods have been determined long before the method is actually applied.

Here are ways to measure class conformity with SRP:

1) Calculate the number of methods in the class, the number of methods is called 'n' or known as the NOM Metric (Number of Method).
2) Collect all types of parameter types that exist in all methods in the class and calculate the number, called 'T'.
3) Then collect the type of parameters used in each method and calculate the amount, referred to as 'P'
4) If all the Metric attributes have been collected, then calculate using the CAMC formula.

$$\boldsymbol{CAMC = \frac{\sum_{i=1}^{n} |Pi|}{|T| x\, n}} \tag{1}$$

5) If the CAMC value is below 0.35, then the class does not meet the SRP, or in this study is called the NCSRP class (Not Conform to SRP), whereas if it is above 0.35 it is called the CSRP (Conform to SRP) class.

Definition of Metrics to assess the compliance of the overall class diagram against the SRP:

$$\boldsymbol{VSRP = \frac{\sum CSRP}{Number\ of\ Class\ (NC)}} \tag{2}$$

VSRP (Value of SRP) has a range of values from 0 to 1. $\sum$CSRP is the number of classes that meet the SRP.

B. Metrics of Open Close Principles (OCP)

Based on literature analysis, this principle is related to inheritance and abstraction. To be open for extendtion, a class must have an inheritance relationship with another class and the baseclass must be in the form of an abstract class, if it meets the two criteria, the class meets the OCP. The measurement ignores the aspect of understanding the domain, so the measured is only on base class, Depth of Inheritance (DIT) = 0, not in all classes, with the consideration of avoiding the OCP value is good but the inheritance structure seems complex.

The following ways to measure class design with OCP:

1) Check DIT for each class. Classes with DIT = 0 is referred to as the Metric Number of Superclass (NSUP)
2) Next, check Number of Children (NOC) and class status. If the NOC > 0 and the class is an abstract class then the

class is considered to have fulfilled the OCP, called COCP (Conform to OCP).

Definition of Metrics to assess the compliance of the overall class diagram against the OCP:

$$\boldsymbol{VOCP} = \frac{\sum \boldsymbol{COCP}}{\boldsymbol{Number\ of\ Superclass\ (NSUP)}} \tag{3}$$

VOCP (Value of OCP) has a range of values from 0 to 1. $\sum$ COCP is the number of classes that meet the OCP.

C. Metrics of Liskov Substitution Principles (LSP)

The main key to this principle is the management of the inheritance hierarchy. The main metrics in this principle are NMI, NMA and NMO. We spesific The NMI metric into NMI^a, the number of abstract methods inherited by the baseclass. NME metrics are the number of abstract methods defined by subclasses, these definitions must be with the same method components (return type, name and parameter) like the baseclass property, whereas NMO metrics are the number of concrete methods overridden and overloading by subclasses. This measurement is done in a set of inheritance hierarchies, all of baseclaas and subclasses pairs must be checked, to the lowest layer.

The following is a technique for measuring class design conformity with LSP:

1) A set of inheritance hierarchies, namely from (DIT = 0) $\wedge$ (NOC > 0) $\wedge$ (abstract) to the largest DIT value at the lowest inheritance level. This set of hierarchies is called the Number of Hierarchies (NOH).
2) This calculation is done one by one between the baseclass inheritance pair and its subclasses in a set of inheritance hierarchies, namely by:
   a. Calculate the number of baseclass/children (NOC).
   b. Then calculate the value of NMI^a baseclass.
   c. Then calculate NME and NMO from each subclass.
   d. Match the value of NME and NMI^a for each pair of baseclass-subclass, make sure the values are the same. Then calculate NMO in the subclass, make sure that the NMO value is 0, if it is conform then it can be stated that the pair meets LSP, CLSP (Conform LSP).
   e. Calculate the NOC value of each subclass, if NOC> 0, then repeat the calculation of step 2 to the children of the subclass.
3) Calculations can stop immediately when encountered NMO > 0, the hierarchy NCLSP (Not Conform LSP).
4) When NME is found to be smaller than NMI^a, then check the subclass properties, if it is abstract, then continue the number 2 iteration to their children, but if the concrete stops, the inheritance pair has subclasses that do not implement abstract methods, the hierarchy NCLSP.
5) A set of inheritance hierarchies is stated to meet LSP when all inheritance pairs fulfill LSP requirements.

Definition of Metrics for assessing the overall compliance of a class diagram against LSP:

$$\boldsymbol{VLSP} = \frac{\sum \boldsymbol{CLSP}}{\boldsymbol{NIP}} \tag{4}$$

VLSP (Value of LSP) has a range of values from 0 to 1. $\sum$ CLSP is the number of inheritance hierarchy sets that meet LSP.

D. Metrics of Interface Segregation Principles (ISP)

This principle is very similar to the first principle, the Single Responsibility Principles (SRP), where a good interface is an interface with a small, cohesive method. And similar to the Liskov Subtitution Principles (LSP), the interface class design cannot force implement classes to use methods that they don't need. Therefore, checking the conformity of the interface class with the ISP simply by reviewing its conformity with SRP and LSP, the interface must meet the SRP and LSP requirements.

Following are ways to measure class conformity with ISP:

1) Check whether the interface class meets the Single Responsibility Principles (SRP)
2) Check whether the interface class is in the inheritance hierarchy that meets the Liskov Subtitution Principles (ISP)
3) When the interface class meets both principles, the class meets the ISP, Conform ISP (CISP).

Definition of Metrics to assess the compliance of an entire class diagram to an ISP:

$$\boldsymbol{VISP} = \frac{\sum \boldsymbol{CISP}}{\boldsymbol{NOI}} \tag{5}$$

VISP (Value of ISP) has a range of values from 0 to 1. $\sum$ CISP is the number of classes that meet the ISP.

E. Metrics of Dependency Inversion Principles (DIP)

This principle confirms that a class may only depend on abstract classes or interfaces. Then the class diagram indication that meets this principle is seen from all classes that have dependencies with other classes compared to the number of classes that depend on abstract classes.

Here are ways to measure class design with DIP:

1) Check the class that has dependency relationships with other classes, when DCC> 0. The total of all DCC in a class diagram, namely NDep Metric (The total number of dependency relationships within a class diagram)
2) If the class has a dependency relationship, check the pair of the class (DCC') whether it is an abstract class or a concrete class. If DCC is an abstract class, the class meets the DIP, CDIP (Conform to DIP) principle.

Definition of Metrics for assessing the overall compliance of a class diagram of a DIP:

$$\boldsymbol{VDIP} = \frac{\sum \boldsymbol{CDIP}}{\boldsymbol{NDep}} \tag{6}$$

VDIP (Value of DIP) has a range of values from 0 to 1. $\sum$ CDIP is the number of classes that meet DIP.

The following is a summary of the proposed criteria for each principle in the form of design properties, OO metrics, conform terms and metrics to calculate the conformity value which can be seen in Table 1. Some of the Metrics that used in this study are 12 Metrics, namely 2 CK metrics [10], 1 Li and Henry's Metrics [11], 2 Bansiya et al .'s metrics [12], 3 Lorenz, 1 Genero et al. It's metrics and 3 other metrics [5].

TABLE 1 LIST OF PROPOSE METRICS

SOLID	Design Properties	OO Metrics	Conform Terms	Metrics
SRP	Class size, cohesion	NOM, CAMC, NC	CAMC > 0,35	$\frac{\sum \boldsymbol{CSRP}}{\boldsymbol{NC}}$
OCP	Inheritance, abstraction	DIT, NOC, NC	(NOC > 0) ∧ (baseclass = abstract)	$\frac{\sum \boldsymbol{COCP}}{\boldsymbol{NC}}$
LSP	Inheritance	NMI, NMO, NIP	NME = NMI^a	$\frac{\sum \boldsymbol{CLSP}}{\boldsymbol{NIP}}$
ISP	Inheritance	NMI, NMO, NOI	CSRP ∧ CLSP	$\frac{\sum \boldsymbol{CISP}}{\boldsymbol{NOI}}$
DIP	copling, abstraction	DCC, DCC', NDep	DCC' = abstract	$\frac{\sum \boldsymbol{CDIP}}{\boldsymbol{NDep}}$

IV. VALIDATION

Validation was carried out by collecting 15 case studies obtained from collecting designs from the assignments of ITB Informatics Masters students and reverse engineering from the open source project taken from the Github website. The measurement of the design conformity value of the case study was calculated by the SOLID Metrics, then compared to the results of the conformity calculation carried out by several experts and practitioners experienced in object-oriented programming. Table 2 is a table of values for the conformity of case studies calculated using SOLID Metrics.

TABLE 2 CONFORMITY VALUE OF 15 CLASS DIAGRAMS WITH SOLID METRICS

Class	Percentage of Conformity (in %)				
	S	O	L	I	D
1	100	33	100	100	0
2	67	100	0	0	0
3	91	13	100	33	40
4	100	100	50	33	100
5	86	50	0	0	0
6	100	33	0	0	100
7	100	67	100	0	100
8	100	100	100	100	100
9	100	20	100	100	0
10	100	100	100	100	100
11	100	100	100	100	100
12	89	50	0	0	0
13	100	50	100	0	0
14	90	20	0	0	67
15	85	57	75	75	71

The following in Table 3 is a table of values for the conformity of case studies calculated using expert judgement.

TABLE 3 CONFORMITY VALUE OF 15 CLASS DIAGRAMS WITH EXPERT JUDGEMENT

Class	Nilai Kesesuaian				
	S	O	L	I	D
1	3.3	2.7	4	3.7	1.7
2	4	4	2	2	1
3	4.3	2.3	4	2.3	1.7
4	4.3	4.3	3	3	3.3
5	4.3	3	1.7	1.3	1.3
6	4.7	2.3	1.7	1	3.3
7	4.7	4.3	4	1	3.3
8	4.7	4.3	4	4.3	4
9	4.7	2	3.3	4	1.3
10	4.7	4.7	4.3	4.3	4
11	4.7	4.7	4.3	4.3	4
12	5	3.7	2.3	2.7	1.3
13	4	3.3	3.3	1.3	1.7
14	4.3	2.3	1.7	1	3
15	3.7	3.7	4.3	4.7	3.3

The hypothesis below is used in the evaluation:

Hypothesis 1

$H_0 : \rho = 0$ There is no significant correlation between SRP Metrics and SRP assessment from experts.

$H_1 : \rho \neq 0$ The SRP metric can predict the conformity value of a class diagram as well as assessment from the expert.

Hypothesis 2

$H_0 : \rho = 0$ There is no significant correlation between OCP Metrics and OCP assessment from experts.

$H_1 : \rho \neq 0$ The OCP metric can predict the conformity value of a class diagram as well as assessment from the expert.

Hypothesis 3

$H_0 : \rho = 0$ There is no significant correlation between LSP Metrics and LSP assessment from experts.

$H_1 : \rho \neq 0$ The LSP metric can predict the conformity value of a class diagram as well as assessment from the expert.

Hypothesis 4

$H_0 : \rho = 0$ There is no significant correlation between ISP Metrics and ISP assessment from experts.

$H_1 : \rho \neq 0$ The ISP metric can predict the conformity value of a class diagram as well as assessment from the expert.

Hypothesis 5

$H_0 : \rho = 0$ There is no significant correlation between DIP Metrics and DIP assessment from experts.

$H_1 : \rho \neq 0$ The DIP metric can predict the conformity value of a class diagram as well as assessment from the expert.

The following is testing of hypothesis with the Spearman's rank correlation using IBM SPSS Statistics Tool. If the value of Sig. < 0.05 Then there is a significant correlation (H_1 Accepted). If the value of Sig. > 0.05 there is no significant correlation (H_0 Accepted).

Hypothesis testing

			Nilai SRP Metrik	Nilai SRP Ahli
Spearman's rho	Nilai SRP Metrik	Correlation Coefficient	1.000	.330
		Sig. (2-tailed)	.	.229
		N	15	15
	Nilai SRP Ahli	Correlation Coefficient	.330	1.000
		Sig. (2-tailed)	.229	.
		N	15	15

Figure 1 Hypothesis 1 Correlation Value

In Figure 2, The value of the correlation is 0.330 meaning that the correlation is low. While the direction of the correlation is in the positive correlation coefficient, it means that if the assessment from the expert is high then the assessment of the metrics is also high. The significance shows the value is 0.229, then H_0 Accepted, meaning that the relationship is not significant between the assessment of the conformity of the class diagram using the SRP Metrics and the assessment of the experts.

			Nilai OCP Metrik	Nilai OCP Ahli
Spearman's rho	Nilai OCP Metrik	Correlation Coefficient	1.000	.941**
		Sig. (2-tailed)	.	.000
		N	15	15
	Nilai OCP Ahli	Correlation Coefficient	.941**	1.000
		Sig. (2-tailed)	.000	.
		N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 2 Hypothesis 2 Correlation Value

In Figure 3, The value of the correlation is 0.941 meaning that the correlation is very strong. The direction of the correlation is in the positive. The significance 0.000, then H_1 Accepted, meaning that the relationship is significant between the assessment of OCP Metrics and the experts.

			Nilai LSP Metrik	Nilai LSP Ahli
Spearman's rho	Nilai LSP Metrik	Correlation Coefficient	1.000	.791**
		Sig. (2-tailed)	.	.000
		N	15	15
	Nilai LSP Ahli	Correlation Coefficient	.791**	1.000
		Sig. (2-tailed)	.000	.
		N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 3 Hypothesis 3 Correlation Value

In Figure 4, The value of the correlation is 0.791 meaning that the correlation is strong. The direction of the correlation is in the positive. The significance 0.000, then H_1 Accepted, meaning that the relationship is significant between the assessment of LSP Metrics and the experts.

			Nilai ISP Metrik	Nilai ISP Ahli
Spearman's rho	Nilai ISP Metrik	Correlation Coefficient	1.000	.880**
		Sig. (2-tailed)	.	.000
		N	15	15
	Nilai ISP Ahli	Correlation Coefficient	.880**	1.000
		Sig. (2-tailed)	.000	.
		N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 4 Hypothesis 4 Correlation Value

In Figure 5, The value of the correlation is 0.880 meaning that the correlation is very strong. The direction of the correlation is in the positive. The significance 0.000, then H_1 Accepted, meaning that the relationship is significant between the assessment of ISP Metrics and the experts.

			Nilai DIP Metrik	Nilai DIP Ahli
Spearman's rho	Nilai DIP Metrik	Correlation Coefficient	1.000	.920**
		Sig. (2-tailed)	.	.000
		N	15	15
	Nilai DIP Ahli	Correlation Coefficient	.920**	1.000
		Sig. (2-tailed)	.000	.
		N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 5 Hypothesis 5 Correlation Value

In Figure 6, The value of the correlation is 0.920 meaning that the correlation is low. The direction of the correlation is in the positive. The significance 0.000, then H_1 Accepted, meaning that the relationship is significant between the assessment of DIP Metrics and the experts.

The following in Table 4 is the relationship between metrics. CC is Correlation Coefficient.

TABLE 4 CORRELATION BETWEEN PRINCIPLES

		SRP	OCP	LSP	ISP	DIP
SRP	CC	1.000	0.099	0.654	0.449	0.467
	Sig.	0.000	0.726	0.008	0.093	0.079
OCP	CC	0.099	1.000	0.077	0.201	0.480
	Sig.	0.726	0.000	0.785	0.473	0.070
LSP	CC	0.654	0.077	1.000	0.674	0.200
	Sig.	0.008	0.785	0.000	0.006	0.475
ISP	CC	0.449	0.201	0.674	1.000	0.256
	Sig.	0.093	0.473	0.006	0.000	0.357
DIP	CC	0.467	0.480	0.200	0.256	1.000
	Sig.	0.079	0.070	0.475	0.357	0.000

From above, there is a correlation between:

1. SRP and LSP metrics

   Correlation value is 0.654, where the value shows a strong correlation. This value also shows a unidirectional relationship, when the value of SRP's metrics rises, LSP's value also increases. In addition, the relationship between metrics is also significant as seen from the significance value of less than 0.05, that is 0.008.
2. LSP and ISP metrics

   Correlation value is 0.674, shows a strong correlation. This value also shows a unidirectional relationship, when the LSP's metrics value increases then ISP's value also rises. In addition, the relationship between metrics is also significant less than 0.05, that is 0.006.

V. DISCUSSION

From the results of metric validation, the confomity of class diagrams with SOLID Metrics produced in this study provides a value that can be used as a consideration for developers in taking policy of their class diagrams design, without assessing it manually and subjectively, so that the resources can be effective for other development processes.

Based on the correlation metric to the experts assessment, it was found that:

1. The correlation value is 'very strong' on OCP's, ISP's and DIP's Metrics
2. LSP's metric correlation values is only 'strong', not as perfect as the previous 3 principles

   It indicated because the class diagram specified writing mismatch.
3. SRP's metric, showing a 'weak' correlation

   Indicated because the value limit of SRP confomity follow the value limit from the CAMC Metric by J. Bansiya, is too low at only 0.35.

There are several factors that can be threat of validation:

1. Sample number of case studies
   With a very small number of case studies, namely 15 case studies, the accidental level of the assessment results is quite large.
2. Subjectivity of experts
   Subjectivity of experts influences validation assessment because the knowledge and point of view of each expert is different.

The results of this study while already showing significant assistance in improving the quality of object-oriented software design.

VI. CONCLUSION

From this research Metrics have been produced along with measurement techniques to measure the conformity of class diagrams on the SOLID Principle. For SRP, VSRP Metric (Value of SRP) is proposed, as well as other principles: OCP - VOCP (Value of OCP), LSP - VLSP (Value of LSP), ISP - VISP (Value of ISP) and DIP - VDIP (Value of DIP).

From the results of the validation it was found that:

1. The conformity of the class diagram design to the SOLID Principle can be measured by Metrics
   4 of 5 metric have a high correlation value to the assessment of experts. Although SRP's Metric has a low correlation value. But this metric as initial research, has been able to help developers to measure the conformity of the design.
2. There is a correlation between Metrics
   The correlation is between SRP metrics with LSP, and LSP with ISP metrics. This correlation shows a unidirectional relationship, where when the value of one metric rises, the value of another metric also rises. This makes it possible to not need to measure the value of conformity.

This research is expected with the value of conformity produced can help developers so that they can be re-checked and prevented as early as possible so as to minimize errors at the next stage of development.

To develop further research several suggestions that can be submitted include:

1. Further development of the conformity metrics of the Single Responsibility Principle (SRP) with other metrics, to obtain a strong correlation value to expert judgment.
2. Adding the number of case studies to get more accurate validation values
3. Involve experts/practitioners with a number and background that corresponds to the number of case studies to support more convincing validation

REFERENCES

[1] R. S. Pressman, *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009.

[2] K. A. Briski *et al.*, "Minimizing code defects to improve software quality and lower development costs .," *Dev. Solut.*, no. October, p. 12, 2008.

[3] R. C. Martin, "Agile Software Development, Principles, Patterns, and Practices," *Book*. pp. 85–145, 2002.

[4] R. Plösch, J. Bräuer, C. Körner, and M. Saft, "MUSE: A Framework for Measuring Object-Oriented Design Quality.," *J. Object Technol.*, vol. 15, no. 4, p. 2:1, 2016.

[5] M. Genero, M. Piattini, and C. Calero, "A survey of metrics for UML class diagrams," *J. Object Technol.*, vol. 4, no. 9, pp. 59–92, 2005.

[6] C. D. and W. L. Jagdish Bansiya, Letha Etzkorn, "A Class Cohesion Metric For Object-Oriented Designs," pp. 1–17, 1998.

[7] R. Plösch, J. Bräuer, C. Körner, and M. Saft, "Measuring, Assessing and Improving Software Quality based on Object-Oriented Design Principles," *Open Comput. Sci.*, vol. 6, no. 1, pp. 187–207, 2016.

[8] E. Chebanyuk and K. Markov, "An approach to class diagrams verification according to SOLID design principles," *Model. 2016 - Proc. 4th Int. Conf. Model. Eng. Softw. Dev.*, pp. 435–441, 2016.

[9] J. Dooley, *Object-Oriented Design Principles*. 2011.

[10] S. R. Chidamber and C. F. Kemerer, "a Metrics Suite for Object Oriented Designa Metrics Suite for Object Oriented Design," *PhD Propos.*, vol. 1, no. 6, pp. 476–493, 1992.

[11] W. Li and S. Henry, "Object-oriented metrics that predict maintainability," *The Journal of Systems and Software*, vol. 23, no. 2. pp. 111–122, 1993.

[12] J. Bansiya and C. G. Davis, "A hierarchical model for object-oriented design quality assessment," *Softw. Eng. IEEE Trans.*, vol. 28, no. 1, pp. 4–17, 2002.