Analysis of Metric Set for Classes in QR Application

S. Bhavani
Assistant Professor (Senior), VIT University, Vellore, India

Abstract— *Software plays critical role in day to day life. Starting from applications like automatic light on/off to complex Roberts, everything needs software for its working. Without Software, computer is just hardware. Even a simple calculator needs quality software to work effectively. The quality of the software is assessed by a collection of software metrics. Often, single metric is insufficient to analyse the software. Instead, the Quality of the software can be determined by a collection of metrics. In this paper, we have analysed the software by a collection of object oriented metrics.*

Keywords— Software Metrics, Metric Set for classes, QR code

I. INTRODUCTION

In object oriented metrics, in addition to metrics which measures the aspects of class, metrics which give details on interaction between classes are also concerned. This approach gives us more information on about our design than our source code. Using object oriented metrics we can measure the static and maintainability of the software. Good software should have low coupling. That is if one class is altered, it should not highly affect other classes. When classes have high level of dependency it should be placed in same package.

A. ***Software Metrics***

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules [1]. Software metrics is used to assess the quality of the software. Fenton & Pfleeger defines software metrics as “the quantitative measure of the degree to which a system, component, or process possess a given attribute" [1]. Initially, metrics like *lines of code (LOC)* and *Number of Methods (NOM)* are concerned. For assessing the staticness of software *Number of Static Methods (NSM)* is used. Dependencies between classes can determined by *Coupling Between Objects (CBO).* Inheritance in software can be tracked using *Number of Overridden Methods (NORM).* We also concerned about Cyclometric Complexity which can be calculated by *Weighted Method per Class (WMC).*

B. ***Quick Response Code(QR Code)***

Quick Response code is the trademark for type of matrix barcode [2]. Barcodes have data only in its rows. Information is stored horizontally in Bar code, whereas in QR code information is stored both horizontally and vertically. This makes QR code to store maximum amount of data compared to Barcode.

Fig. 1 Bar code and QR code

Information can be retrieved even if QR image is partially damaged. The read soloman error correction is used to recover the data. Based on different error correction values the different versions of QR code are available. The Numeric, alphabetic, Kanji, Kana characters can be stored in the QR code. Especially, the alphabetic characters version 40 could be stored for 4,296 characters. Internationally, many standards accept QR codes, such as ISO/IEC 18004:2000[3] and ISO/IEC 18004:2006[4].

QR code Scanner uses the camera on smartphone and look up for information like web sites hyperlinks, contact information like phone numbers and email address, map locations and many more. In simple, QR code can be said as ‘image based hypertext link. QR code is widely used in Advertising, URLs, Virtual stores, Code Payments, Website Login. The QR code's versatility is the main advantage. QR codes can be used for all purpose. Using QR code is beneficial for both customers and business. QR code can be printed on anything. They can be on magazines, billboards, even on coffee cup enticing to scan [5].

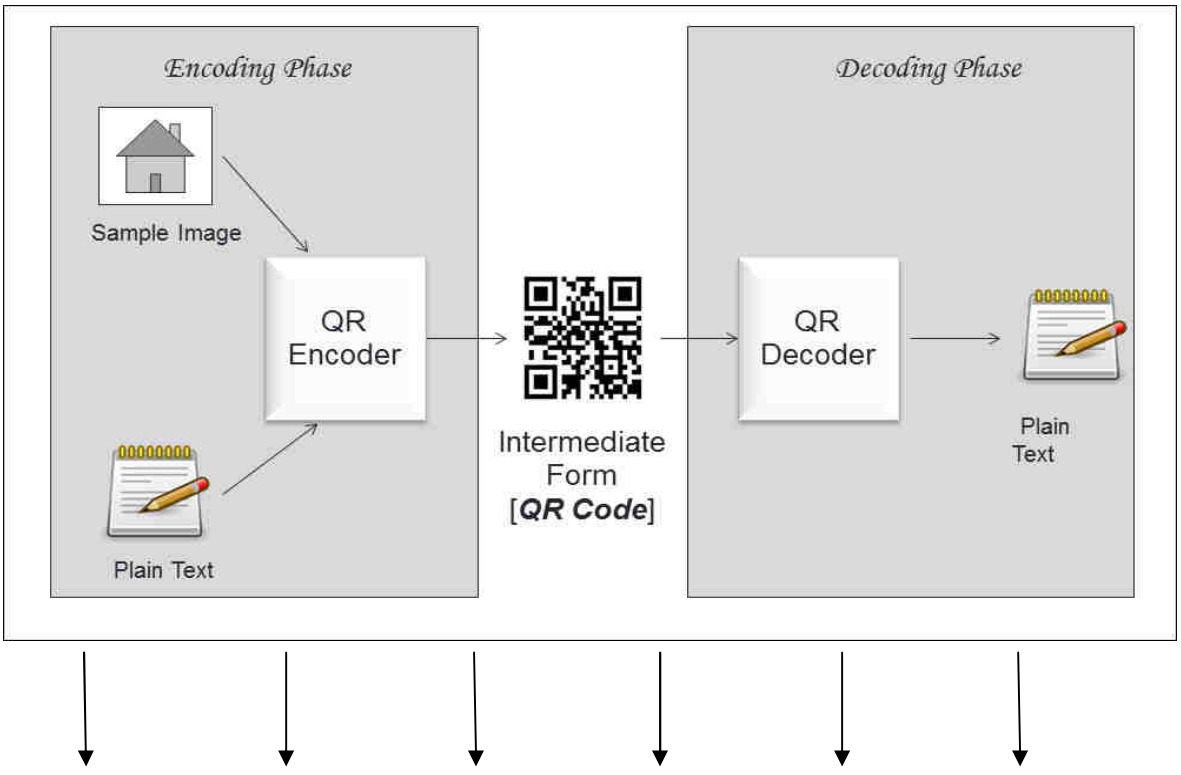
Fig. 2 Versatile Nature of QR code

II. SYSTEM ARCHITECTURE

This paper aims to minimize the metric set of object oriented software Application (QR Application developed in java). Figure 3 shows the overall architecture of this project. QR application contains QR Encoder which encodes plain text to QR image and QR Decoder which retrieves information in QR code [6]. Metrics are calculated for this complete QR application and optimized using thresholds.

1. Quick Response Code

As a name says Quick Response code helps to get quickly responded as soon as it is scanned. Almost every one of us now own smartphone. QR code is easy way for communicating between customers and suppliers.



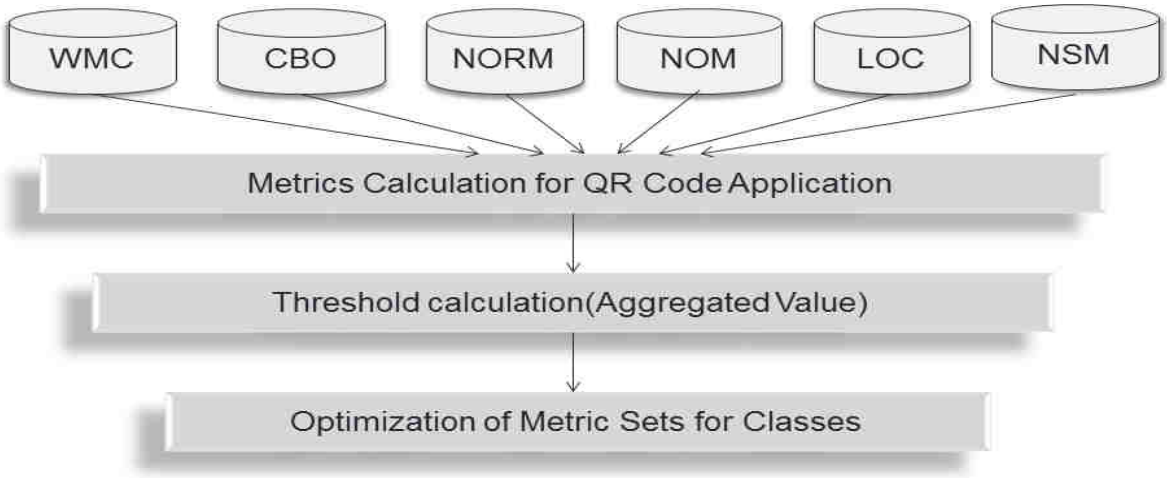


Fig. 3 System Architecture

1.1 QR Encoder

QR Encoder/QR Generator is an application which takes data and image as input and generates QR code [7]. To put in a simple terms, QR Generator is a used as pattern generator in QR code. Without using QR Scanner/QR Decoder, information cannot be retrieved from QR code.

1.2 QR Decoder

QR Decoder/QR Scanner is an application which retrieves information from given QR code. It is a Decoding program which decodes the Intermediate form(which is in form of QR code) to plain text.

2. Metrics Calculation

Often, single metric does not cover all the aspects of software; we have used a collection of software metrics that covers attributes like size, structure and complexity.

2.1 Metrics Calculation of QR Application

The collection of metrics listed in Table 1 is used to analyze the classes. Along with these metrics, five internal attributes of classes are evaluated. The **Weighted Methods per Class** (WMC) is the sum of Encoding program which converts plain text to intermediate form which is in form of Quick Response code with the help of dummy image [8]. The input image has nothing to do with actual content. It is only

complexities of all class methods. The number of redefined operations plays a role in the specialization of the class and must be maintained in a proportion that continues to justify inheritance [9]. These operations are maintained by the metric **Number of Overridden Methods** (NORM) [9]. Size of classes can be calculated using **Number of Methods** (NOM) and **Lines of Code** (LOC) [9]. Static Property is main advantage of Object oriented programming and it can be measured using **Number of Static Methods** (NSM) [9]. Coupling is measured using **Coupling Between Objects** (CBO). Use links between classes define the detailed architecture of the application, just as use links between packages define the highest level architecture to calculate CBO [10].

$$CBO = \frac{NumberOfLinks}{NumberOfClasses} \quad \text{--------> (1)}$$

Metric Name	Internal Attribute	Description
Weighted Methods per Class(WMC)	Method Complexity	Complexity of a class as the sum of the complexity of all methods in the class. Here, VG is used as complexity measure.

Coupling Between Objects(CBO)	Coupling	Number of classes, a class is coupled to.
Number of Overridden Methods(NORM)	Inheritance	Number of methods defined by a parent that are overridden by a class.
Number of Methods(NOM)	Size	Number of methods in a class
Lines of Code(LOC)	Size	Lines of code without considering empty and comment-only lines
Number of Static Methods(NSM)	Staticness	Number of static methods in a class

Table 1 Metric Set under study

2.2 Threshold Calculation

Maximum value, Minimum value, Arithmetic Mean, Median values are calculated for all the input files. Using these values Threshold or Aggregated value is calculated using following simple formula,

T=Mean(a,b,c)+[Max(a,b,c) – Min(a,b,c)] / 2 ------(2)

Where

T- Threshold or Aggregated value

a,b,c – Metric values of input files

2.3 Analysis of Metric Set

The calculated aggregated value is checked with optimal values of the Metrics that is determined for the environment of Application developed. If the aggregated value lies between optimal range of that metric, it is taken for final result set [11].

Metric Name	Optimal Range
WMC	1-50
CBO	1-4
NORM	0-5
NOM	3-7
LOC	NA
NSM	1-10

Table 2 Optimal Range for Metric Set

III.RESULTS AND DISCUSSIONS

Clear indicators are required to determine appropriateness of metric values. Hence, we use thresholds for metric values. For the analysis of classes, the six metrics {WMC, CBO, NORM, NOM, LOC, NSM) calculated for complete QR Application which includes QR Encoder, QR Decoder and Main Form. Maximum value, Minimum value, Arithmetic Mean, Median values are calculated for all the input files. Using these values, aggregated value for each metric is calculated. And these values are compared with optimal limits (Table 2) of their respective metrics.

Metric	Encoder	Decoder	Main Form	Mean	Median	Max	Aggregated
LOC	115	168	145	145	142.67	168	169
NOM	10	7	5	7	7.33	10	9
NSM	1	1	1	1	1.00	1	1
CBO	7	6	6	6	6.33	7	6

NORM	0	0	0	0	0.00	0	0
WMC	9	9	5	9	7.67	9	9

Table 3 Statistical information about the measured projects

Comparing Aggregated values of metrics with its optimal range which listed in Table 2, the optimized metric set is M= {WMC, NORM, NSM}. Calculated CBO value is larger than its optimal value which means the QR Application project is loosely coupled. To make it tightly coupled, the project source code should be divided into smaller files.

REFERENCES

[1] Fenton, N.E., & Pfleeger, S.L. (2004). Software Metrics: A Rigorous & Practical Approach (2nd ed. Revisited Printing). London: International Thomson Computer Press.

[2] QR code, http://www.qrcode.com/en/.

[3] ISO/IEC 18004:2000(E), "Information technology Automatic identification and data capture techniques Bar code symbology QR Code", 2000.

[4] ISO/IEC 18004:2006(E), "Information technology. Automatic identification and data capture techniques. QR Code 2005 bar code symbology specification" ,2006.

[5] Yuji Kato, Daisuke Deguchi, Tomokazu Takahashi, Ichiro Ide and Hiroshi Murase "Low resolution QR-code recognition by applying super-resolution using the property of QR-codes*" 2011 International Conference on Document Analysis and Recognition*

[6] Yu-Hsun Lin, Yu-Pei Chang, and Ja-Ling Wu "Appearance-Based QR Code Beautifier" *IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 15, NO. 8, DECEMBER 2013*

[7] Jitesh .B. Dundas "Understanding Code Patterns Analysis, Interpretation & Measurement" *2008 International Conference on Computer and Electrical Engineering*

[8] XiaoYouan, Yang Chao and Luo Chunling "A new Method of QR code Accumulation Encoding in Mobile Education "*2011 IEEE*

[9] Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki Fukazawa "A Metrics Suite for Measuring Reusability of Software Components" *2003 IEEE*.

[10] http://support.objecteering.com/objecteering6.1/help/us/metrics/metrics_in_detail/coupling_between_object_classes.htm

[11] Paulo Meirelles, Carlos Santos Jr., João Miranda, Fabio Kon, Antonio Terceiro, Christina Chavez "A Study of the Relationships between Source Code Metrics and Attractiveness in Free Software Projects".