**Project preparation/proposal**

**1. Team Information**

**Team I**

|  |  |  |
| --- | --- | --- |
| **Team Member Names** | **Student ID** | **Contact Email** |
| Chetan Paliwal |  |  |
| Himen |  |  |
| Sandeep Siddaramaiah | 40087428 | sandeepsiddaramaiah@gmail.com |
| Karthik |  |  |
| Rohan |  |  |

**2. Selected Metrics and Correlation analysis**

**Metric 1 :**

**Metric 2 :**

**Metric 3 :**

**Metric 4 :**

**Metric 5:**

**Metric 6 : QMOOD – Quality Metrics in Object Oriented Design**

|  |  |
| --- | --- |
| **Quality Factor** | **Definition** |
|  |  |
| Reusability | Shows how easily a design can be applied to a new problem without |
|  | significant efforts provided that there are object oriented design |
|  | characteristics present in the system. |
|  |  |
| Flexibility | Shows how easily changes can be incorporated in a design. |
|  | The ability of a design to be adapted to provide functionality related |
|  | capabilities. |
|  |  |
| Understandability | Shows how easily properties of designs can be learned and |
|  | comprehended. Understandability relates directly to the complexity of |
|  | design structure. |
|  |  |
| Functionality | This is the responsibility that has been assigned to the classes of a |
|  | Design, which can be accessed through the public interfaces. |
|  |  |
| Extendibility | Refers to their presence and usage of properties in an existing design |
|  | that allow for the incorporation of new requirements in the design. |
|  |  |
| Effectiveness | This shows how the Object Oriented Design concepts have been |
|  | included in the design to achieve the desired functionality and |
|  | behaviour. |

**Definition of QMOOD design properties**

|  |  |
| --- | --- |
| **Design Property** | **Definition** |
|  |  |
| Design Size (DSC) | A measure of number of classes used in the design. |
|  |  |
| Hierarchies | Hierarchies are used to represent different generalization – |
| (NOH) | specialization aspects of the design. Classes in a design which have |
|  | one or more descendants exhibit this property. |
|  |  |
| Abstraction (ANA) | A measure of generalization- specialization aspect of design. Classes |
|  | in a design which have one or more descendent exhibit this property of |
|  | abstraction. |
|  |  |
| Encapsulation | Defined as the enclosing of data and behavior within a single |
| (DAM) | construct. In object oriented designs the property specifically refers to |
|  | designing classes that prevent access to attribute declarations by |
|  | defining them to be private, thus protecting the internal representation |
|  | of the objects. |
|  |  |
| Coupling (DCC) | Defines the inter dependency of an object on other objects in a design. |
|  | It is the measure of the number of other objects that would be accessed |
|  | by an object in order for that object to function correctly. |
|  |  |
| Cohesion (CAM) | Accesses the relatedness of methods and attributes in a class. Strong |
|  | overlap in method parameters and attribute types is an indication of |
|  | strong cohesion. |
|  |  |
| Composition | Measures the “part-of,” “has”, “consists –of”, or “part-whole” |
| (MOA) | relationships, which are aggregation relationships in object oriented |
|  | design. |
|  |  |
| Inheritance (MFA) | A measure of the “is-a” relationship between classes. This relationship |
|  | is related to a level of nesting of classes in an inheritance hierarchy. |
|  |  |
| Polymorphism  (NOP) | |  | | --- | | The ability to substitute objects whose interfaces match for one | | another at runtime. It is a measure of services that are dynamically | | determined at run-time in an object. | |
| Messaging (CIS) | |  | | --- | | A count of number of public methods that are available as services to | | other classes. This is the measure of the services that a class provides. | |
| Complexity | A measure of the degree of difficulty in understanding and |
| (NOM) | comprehending the internal and external structure of classes and their |
|  | relationships. |
|  |  |

**QMOOD Quality Factors and Design Properties Relationships**

|  |  |
| --- | --- |
| **Quality Factor** | **Relationship** |
|  |  |
| Reusability | -0.25\*Coupling+0.25\*Cohesion +0.5\*Messaging +0.5\*Design Size |
|  |  |
| Flexibility | 0.25\*Encapsulation-0.25\*Coupling +0.5\*Composition |
|  | +0.5\*Polymorphism |
|  |  |
| Understandability | -0.33\*Abstraction+0.33\*Encapsulation- |
|  | 0.33\*Coupling+0.33\*Cohesion -0.33\*Polymorphism– |
|  | 0.33\*Complexity –0.33\* Design Size |
|  |  |
| Functionality | 0.12\*Cohesion+0.22\*Polymorphism +0.22\*Messaging+ 0.22\*Design |
|  | Size +0.22\*Hierarchies |
|  |  |
| Extendibility | 0.5\*Abstraction– 0.5\*Coupling+0.5\*Inheritance +0.5\* Polymorphism |
|  |  |
| Effectiveness | 0.2\*Abstraction+0.2\*Encapsulation |
|  | +0.2\*Composition+0.2\*Inheritance +0.2\*Polymorphism |
|  |  |

**3.Related Work**

**[6]QMOOD metric sets to assess quality of java program**

**Objective:**

Here a model to evaluate and grade java programs, based on QMOOD which is hierarchical model that defines relation between qualities attributes and design properties with the help of equations. This paper focuses on only MOOD and QMOOD. Different types of java programs are shown as input and result have been evaluated and featured with the help of 2D graph. ISO/lEC 9126 is one of the most popular quality standards.

**Hypothesis:**

The methodology used in the development of hierarchical QMOOD assessment extends Dromey's generic quality model methodology and involves the four levels(Ll through L4) and three mappings(Mapping Quality, Assigning design metrics to design properties and Linking design Properties to Quality Attributes).

**Design Metrics**:

Design size(DSC)- It measures the number of classes used in design

Hierarchies(NOH)- Number of class hierarchies is represented as number of root classes in class design.

Abstraction (ANA) - it measures generalization and specialization in class design.

Encapsulation (DAM) - an enclosure of data and behaviour within single construct.

Coupling (DCC) - Interdependency of an object on other object in a design.

Cohesion (CAM) - It assesses how attributes and methods are related.

Composition (MOA) - Measure of aggregation relationship.

Inheritance (MFA)-Measure of IS-A relationship.

Polymorphism (NOP)-Ability to take different forms in a design.

Messaging (CIS)-Measure of services that a class provides.

Complexity (NOM)-Represents how difficult understand and comprehend the internal and external structure of classes and relationships.

**Methodology:**

From the design quality attribute values, the total quality index was calculated as:

Input: java program to be evaluated.

Output: evaluation results.

Step 1: Compute all the design metrics from the java program taken as input

Step 2: Normalize all the values of the design metrics.

Step 3: Find all the design properties from the design metrics

Step 4: Compute all the quality attributes from the formulas

Step 5: Add the values of all quality attributes to compute the total quality index.

**Results:**

-Design that was having single class has lowest TQI and as the design metrics like design size, inheritance, abstraction, inheritance is increased TQI is increased but it can also be seen that decrease in complexity increase the TQI.

* As the design metrics change TQI varies. As the classes and other properties are included TQI is increased. Further when more private attributes are included that is DAM IS included there is further increase in TQI.

-Coupling and complexity are inversely proportional to the quality of software while others to

some extent give the positive result that is directly proportional to quality.

-The system concludes that it's not necessary that every time increasing the positive parameter would result in better quality.

**4. Projects**

Below is the list of projects among which 5 Projects will be considered for our Analysis purposes.

|  |  |
| --- | --- |
| **Projects** | **Version** |
|  |  |
| Apache Ant | 1.10.5 |
|  |  |
| JRuby | 9.2.5.0 |
|  |  |
| JMeter | 5.0 |
|  |  |
| JFreeChart | 1.0.19 |
|  |  |
| BlueJ | 4.1.4 |
|  |  |
| Mockito | 2.23.20 |
|  |  |

These are open source software systems which are built using Java programming language. These projects fall under different ecosystems as Apache Ant is a software tool for automating software build processes, JRuby is a software used for development of Ruby programming language, JMeter is software used as a load testing tool for analyzing and measuring performance of a variety of services, Jfreechart is used for creation of wide variety of both interactive and non interactive charts, BlueJ is a IDE(integrated development environment) for the Java programming language, developed mainly for educational purposes, but also suitable for small-scale software development and Mockito is a testing framework used for the purpose of test-driven development or behavior-driven development.

These projects have high volume of users and commits. Several developers have worked on these projects which provide a good base to start our analysis on different metrics which we have studied. These projects have several bugs and features reported which provide insight to understand the metrics. They have many version releases which helps to understand the system better with stable builds.

**6. Resource Planning.**

**7. References.**