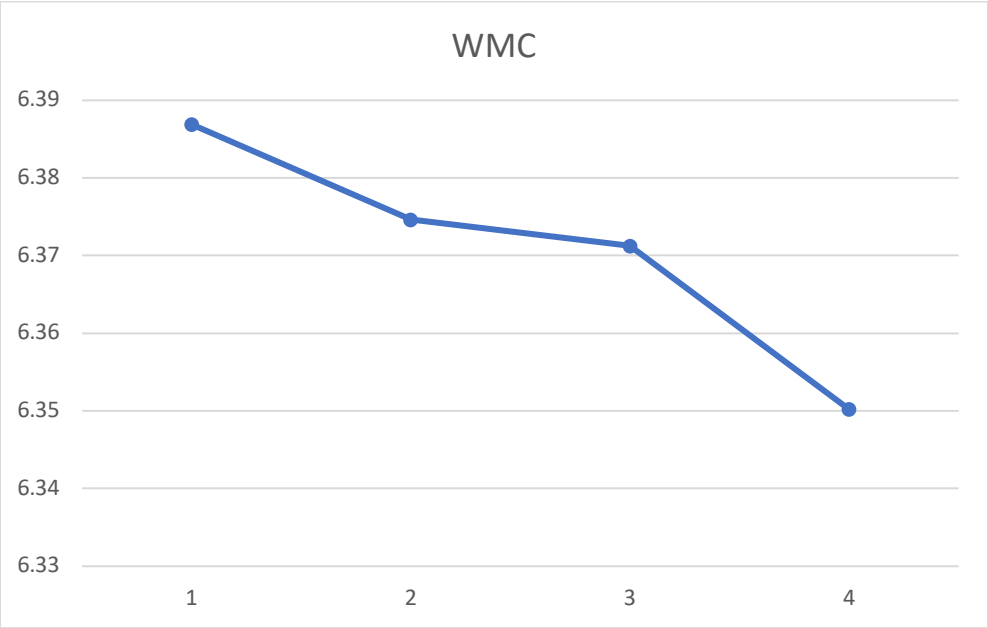
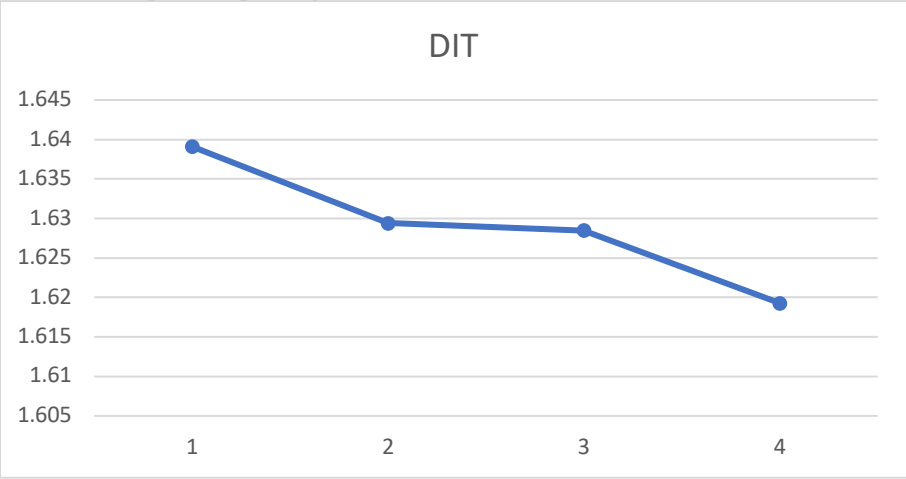
Weighted Methods per class (WMC)

This graph is extracted from apache tomcat versions this graph shows average of each version, however the first version have more complexity and then it reduced in each version, however I suggest to reduce complexity of each method so it can be easily maintainable and as value of WMC increase it will take more efforts and time to maintain class.



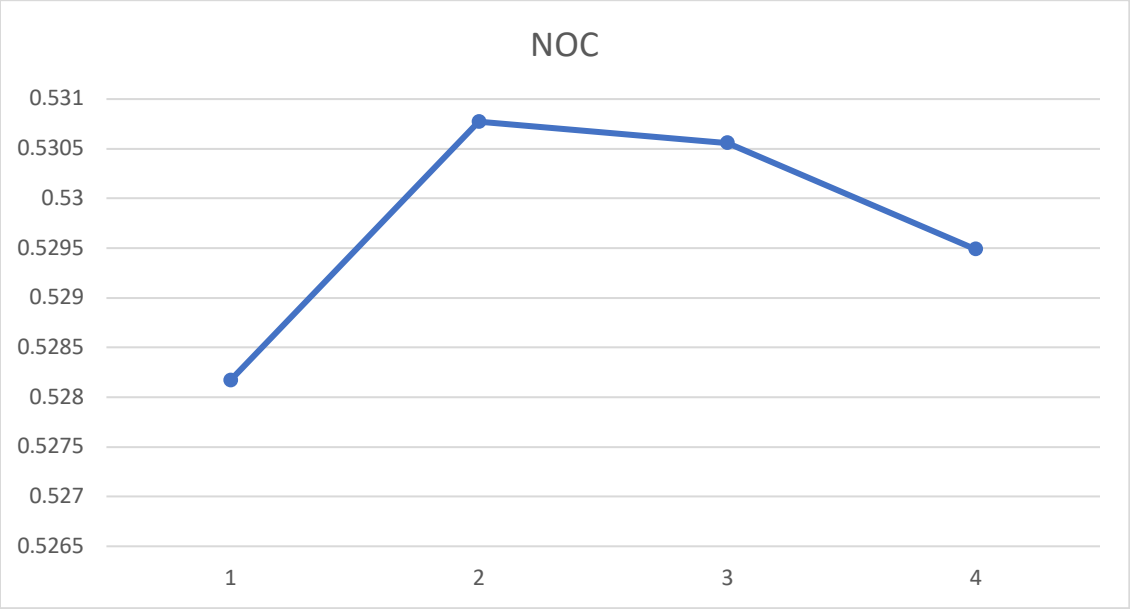
Depth of Inheritance Tree (DIT)

This graph shows fluctuation in depth of inheritance in each version, developers of this software has improve depth of inheritance tree by reducing the path between parent class to root class, my suggestion for developers that reduce path from parent class to root class as it will reduce complexity and easy to understand and locate the methods and classes, moreover it will also reduce design complexity.



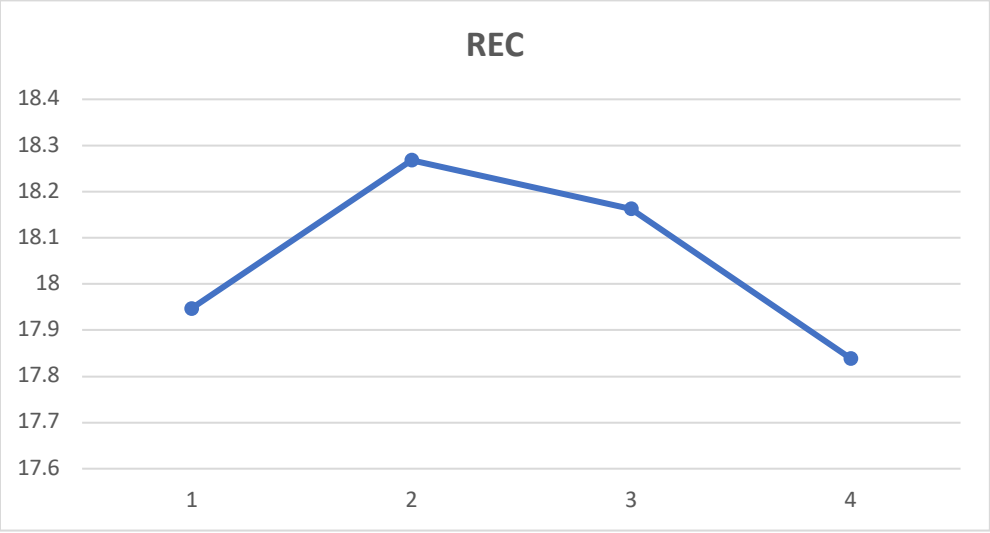
Number of Children (NOC)

This graph shows number of direct subclasses of main class, developers should maintain number of subclasses because it should be easy to abstract parent class, however developers are advised to reduce subclasses in next version to make this value moderate, so it will allow reusability.



Response For a Class (RFC)

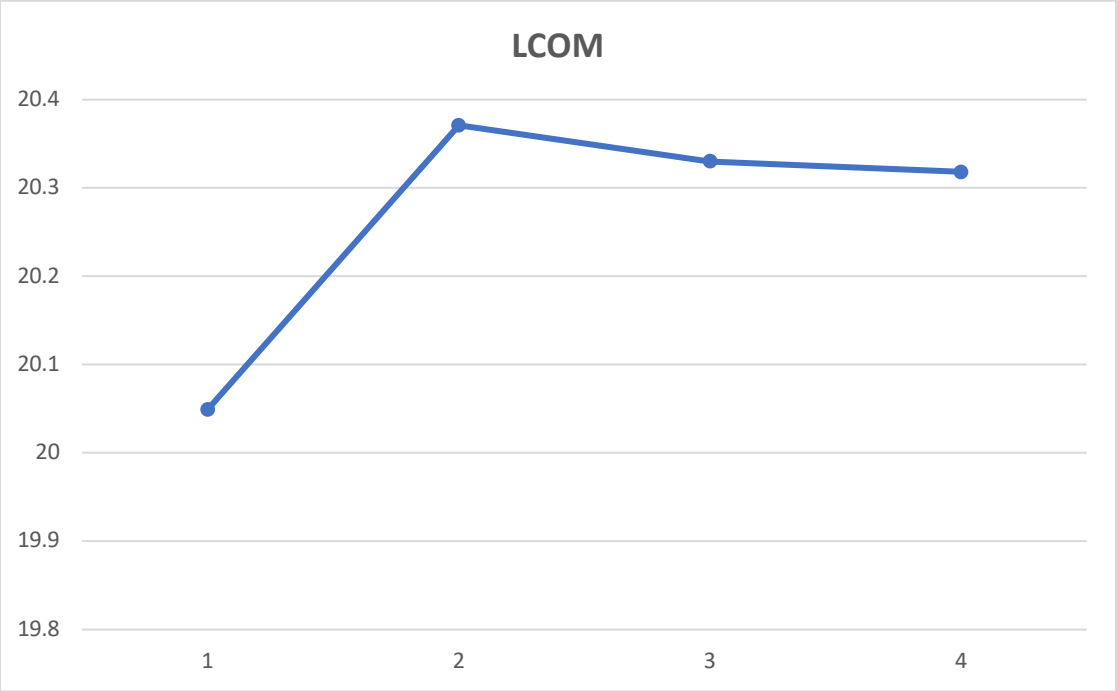
This graph shows number of methods and complexity of classes in each version, so my suggestion is to reduce number of response for a class matrices by reducing number of method invoked by using other object so testing and debugging of class can be done easily.



Coupling Between Objects (CBO)

This graph shows coupling between objects in each versions so coupling is relationship between classes through object so objects are connected with each other, my suggestion for developers of this software to reduce the connection between classes in next version to reduce the chances of faults that might be caused by inter-class activity.

Lack of Cohesion of Methods (LCOM)
This graph is showing average of lack of cohesion of methods in four versions of apache Tomcat, so my suggestion for developers for next version of apache tomcat to reduce the amount of similarity of each method in class so it will be reduce cohesion, to reduce LCOM use different method with different operations using different variables it will reduce the number of lack of cohesion of method.



Examples of CK-Matrices

- **Weighted Methods per Class (WMC)**
  for calculating weighted method of class I have choose public class websocket.drawboard.wsmessages.StringWebsocketMessage on line number 39362 version 8.5.7 having 2 methods that are websocket.drawboard.wsmessages.StringWebsocketMessage.StringWebsocketMessage, websocket.drawboard.wsmessages.StringWebsocketMessage.getString

  WMC = 2

- **Depth of Inheritance Tree (DIT)**
  for calculating depth of inheritance tree which means we have to calculate path from class to root class, line# 39265 version 8.5.7 public class websocket.chat.ChatAnnotation length of this class from node to root is 1 websocket.chat => ChatAnnotation

  DIT=1

- **Number of Children (NOC)**
  This Matrices calculate number of children of class (immediate children) of that class, I have choose line number 38832 version 8.5.7 public static abstract class the node is org.apache.tomcat.websocket.server.TestClose and child is org.apache.tomcat.websocket.server.TestClose.BaseEndpointConfig

  NOC = 1

- **Response for a Class (RFC)**
  This matrices is calculated by counting methods in the class and methods invoked from an other class with other object, so I choose line number 38821 version 8.5.7 Tomcat Public static class org.apache.tomcat.websocket.server.TestClassLoader.ClassLoaderEndpoint having 2 methods org.apache.tomcat.websocket.server.TestClassLoader.ClassLoaderEndpoint.onMessage and org.apache.tomcat.websocket.server.TestClassLoader.ClassLoaderEndpoint.onOpen

  RFC = 2

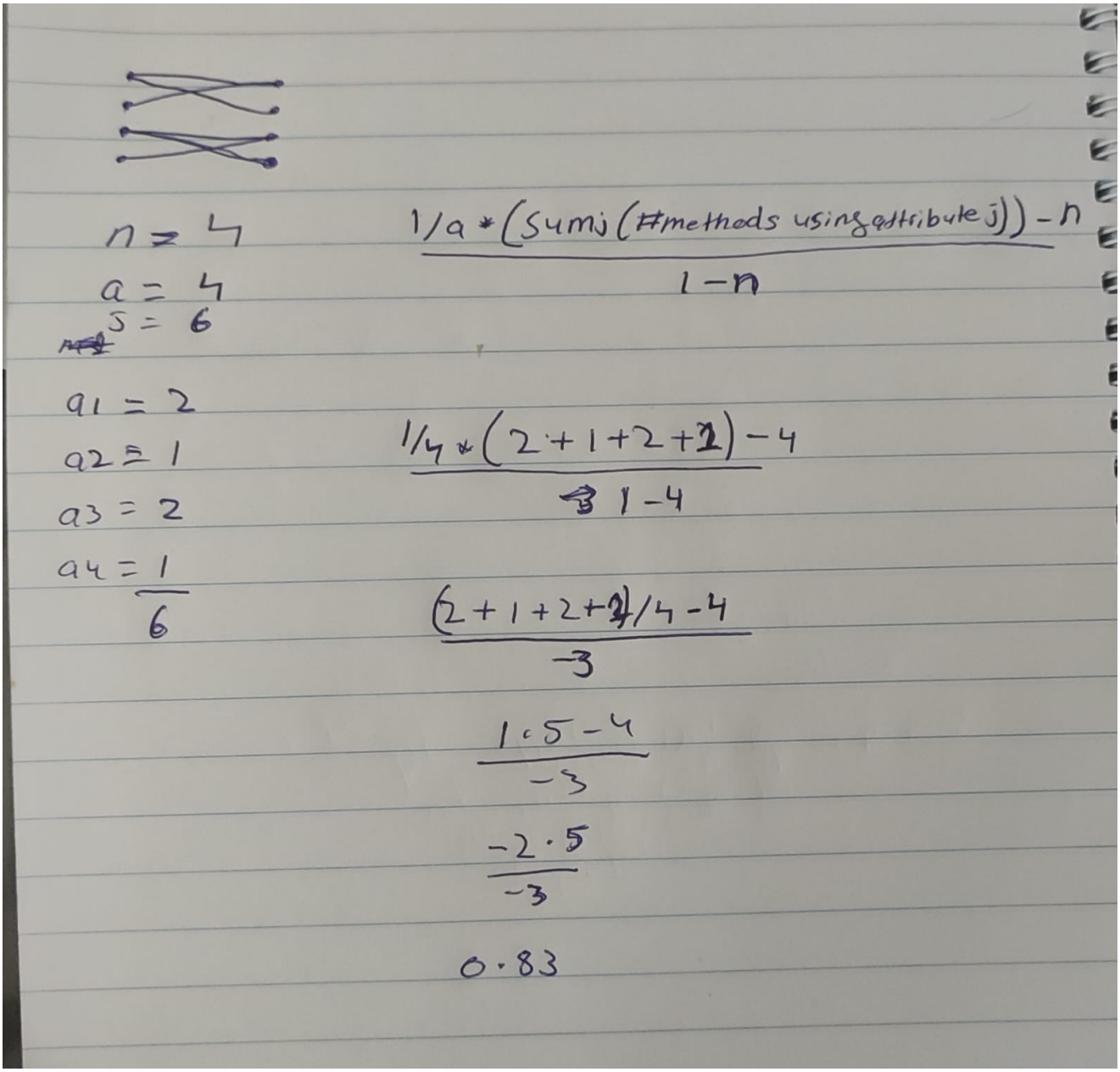
- **Coupling Between Objects (CBO)**
  This matrices is calculated by number of objects that are connected with each other in classes or counting number of other classes that are couple with other.
  I have choose line number 38801 version 8.5.7 public class org.apache.tomcat.websocket.pojo.TesterUtil and line number 38802 public static class org.apache.tomcat.websocket.pojo.TesterUtil.ServerConfigListener so it means it have 1 coupling class

  CBO = 1

- **Lack of Cohesion of Methods (LCOM)**
  This matrices calculated by number of methods using same variables to perform operations. I choose line number 37686 version 8.5.7 public class org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure
  It have 4 methods using same variables

Methods:
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.read
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.read
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.write
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.write

Variables:
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.read.A
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.read.B
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.write.A
org.apache.tomcat.websocket.AsyncChannelWrapperNonSecure.write.B



LCOM = 0.83