[cover sheet 1 of 2]

2

ASSIGNMENT 2

**DEFECT ANALYSIS REPORT**

CSE 6329 -- SOFTWARE MEASUREMENT AND QUALITY ENGINEERING

Professor Dennis J. Frailey

**Fall, 2017**

NAME \_\_\_**Ridham Kothari**\_\_\_

ID Number\_\_\_**1001446086**\_\_\_

[cover sheet 2 of 2]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Grading template Student do not write inside this box** | | | | | | | |
| \_\_\_\_\_ (/16) | 1.0 Description of Analysis Tool (spreadsheet)  1.1 \_\_\_\_ Purpose (1) 1.2 \_\_\_\_ Data Spreadsheet (3) 1.4 \_\_\_\_ Analysis Tool (12) | | | | | | (16 points) |
|  | 2.1 Details for each of the Post Release Quality graphs | | | | | | |
| \_\_\_\_\_\_\_ (/2) | 2.1.1 Overview of Post Release Quality (PRQ) graphs (2) | | | | | | |
|  | 2.1.2 … 2.1.7 Details of Individual Post Release Quality graphs (10 points each) | | | | | | |
| Introduction & Graph  (2.1.x), (2.1.x.1)  (2 points) | Analysis & Discussion  (2.1.x.2)  (2 points) | Procedure Used to Produce the Graph (2.1.x.3) | | | | |
| Base Metrics Collected  (2.1.x.3.1)  (1 point) | Compound Metrics  (2.1.x.3.2)  (1 point) | Data Refinement  (2.1.x.3.3)  (2 points) | How to Interpret  (2.1.x.3.4)  (2 points) | |
| Example  Product ZD  \_\_\_\_\_ (/10) | (2.1.2.1)  \_\_\_\_\_ | (2.1.2.2)  \_\_\_\_\_ | (2.1.2.3.1)  \_\_\_\_\_ | (2.1.2.3.2)  \_\_\_\_\_ | (2.1.2.3.3)  \_\_\_\_\_ | (2.1.2.3.4)  \_\_\_\_\_ | |
| Average  \_\_\_\_\_ (/10) | (2.1.3.1)  \_\_\_\_\_ | (2.1.3.2)  \_\_\_\_\_ | (2.1.3.3.1)  \_\_\_\_\_ | (2.1.3.3.2)  \_\_\_\_\_ | (2.1.3.3.3)  \_\_\_\_\_ | (2.1.3.3.4)  \_\_\_\_\_ | |
| Normalized by Size  \_\_\_\_\_ (/10) | (2.1.4.1)  \_\_\_\_\_ | (2.1.4.2)  \_\_\_\_\_ | (2.1.4.3.1)  \_\_\_\_\_ | (2.1.4.3.2)  \_\_\_\_\_ | (2.1.4.3.3)  \_\_\_\_\_ | (2.1.4.3.4)  \_\_\_\_\_ | |
| PRQ By  Process  \_\_\_\_\_ (/10) | (2.1.5.1)  \_\_\_\_\_ | (2.1.5.2)  \_\_\_\_\_ | (2.1.5.3.1)  \_\_\_\_\_ | (2.1.5.3.2)  \_\_\_\_\_ | (2.1.5.3.3)  \_\_\_\_\_ | (2.1.5.3.4)  \_\_\_\_\_ | |
| PRQ By Language  \_\_\_\_\_ (/10) | (2.1.6.1)  \_\_\_\_\_ | (2.1.6.2)  \_\_\_\_\_ | (2.1.6.3.1)  \_\_\_\_\_ | (2.1.6.3.2)  \_\_\_\_\_ | (2.1.6.3.3)  \_\_\_\_\_ | (2.1.6.3.4)  \_\_\_\_\_ | |
| Student’s Choice  \_\_\_\_\_ (/10) | (2.1.7.1)  \_\_\_\_\_ | (2.1.7.2)  \_\_\_\_\_ | (2.1.7.3.1)  \_\_\_\_\_ | (2.1.7.3.2)  \_\_\_\_\_ | (2.1.7.3.3)  \_\_\_\_\_ | (2.1.7.3.4)  \_\_\_\_\_ | |
|  | 2.2 Details for each of the Post Release Quality History graphs | | | | | | |
| \_\_\_\_\_\_\_ (/2) | 2.2.1 Overview of Post Release Quality History (PRQH) graphs (2) | | | | | | |
|  | 2.2.2, 2.2.3 Details of Individual Post Release Quality graphs (10 points each) | | | | | | |
| Graph  (2.2.x 1)  (2 points) | Analysis & Discussion  (2.2.x.2)  (2 points) | Procedure Used to Produce the Graph (2.2.x.3) | | | | |
| Base Metrics Collected  (2.2.x.3.1)  (1 point) | Compound Metrics  (2.2.x.3.2)  (1 point) | Data Refinement  (2.2.x.3.3)  (2 points) | How to Interpret  (2.2.x.3.4)  (2 points) | |
| PRQH by Quarter  (2.2.2)  \_\_\_\_\_ (/10) | (2.2.2.1)  \_\_\_\_\_ | (2.2.2.2)  \_\_\_\_\_ | (2.2.2.3.1)  \_\_\_\_\_ | (2.2.2.3.2)  \_\_\_\_\_ | (2.2.2.3.3)  \_\_\_\_\_ | (2.2.2.3.4)  \_\_\_\_\_ | |
| PRQH by Year  (2.2.3)  \_\_\_\_\_ (/10) | (2.2.3.1)  \_\_\_\_\_ | (2.2.3.2)  \_\_\_\_\_ | (2.2.3.3.1)  \_\_\_\_\_ | (2.2.3.3.2)  \_\_\_\_\_ | (2.2.3.3.3)  \_\_\_\_\_ | (2.2.3.3.4)  \_\_\_\_\_ | |
| \_\_\_\_\_\_\_  (/100) | Total Assignment Grade | | | | | | |

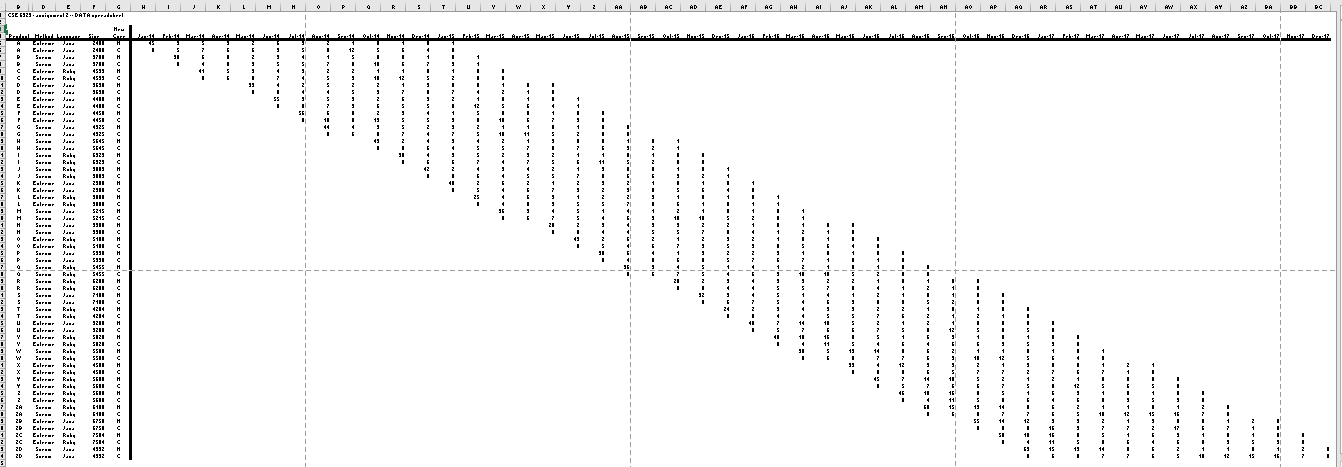
**Defect Analysis Report**

1. **Introduction**
   1. **Purpose of This Report**

This report shows the results of analyzing three years of defect data on our released products. The purpose is to gain a greater understanding of the quality levels of our released products and to determine whether there is any correlation between software quality and other factors such as the programming language used, the development process used, or the time when the product was developed.

* 1. **Data Used**

The data necessary to perform these measurements have been collected monthly for each active software product, over the past three years. The data are stored in the **data spreadsheet**, named A2data.xlsx. The data spreadsheet looks as follows



The above spreadsheet has two rows for each product. The first row ("N") is the number of new defects detected during the month and the second row ("C") is the number of defects corrected during the month.

* **Rel Date** is the date when the product was released. Note that the figure above shows different dates from the ones in your data spreadsheet.
* **Product** is the name assigned to the product during development (a different and more descriptive name is assigned when the product is sold to customers)
* **Method** is the name of the development process (lifecycle) used
* **Language** is the name of the programming language used to write the software. Note that the languages in your data spreadsheet may differ from those shown above.
* **Size** is in source lines of code
* **N** means that this row contains the number of new defects detected during the month
* **C** means that this row contains the number of defects corrected during the month
  1. **Analyses and Graphs**

We have analyzed the data in several different ways, resulting in two different metrics and their corresponding graphs:

1. Post Release Quality (shown six ways),
2. Post Release Quality History (shown two ways).

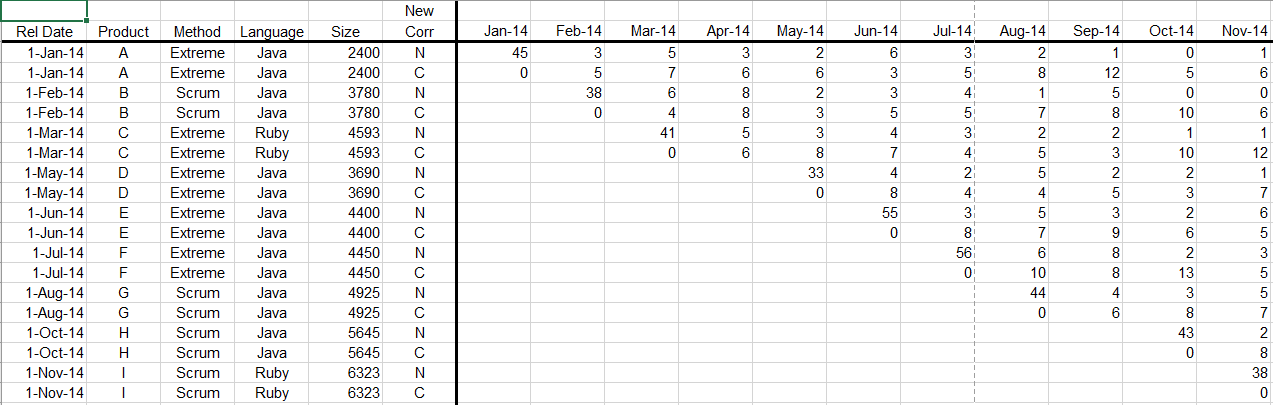
These are described in further detail in section 2 of this report.

* 1. **Structure of Analysis Tool**

In order to analyze defect data, we have created a workbook that contains 7 worksheets in which 1 worksheet contains raw data and other 6 worksheets contains graphs. Each worksheet is described below.

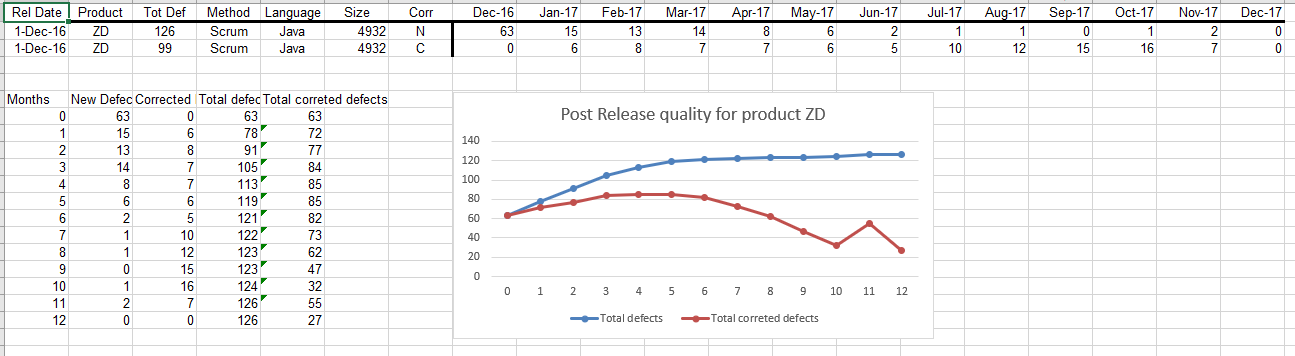
* **Worksheet1:** Data worksheet

This worksheet contains data of products named as A-ZD total 30, with the release data, method used, programming language used, size of the code, new and corrected defects each month for each product.



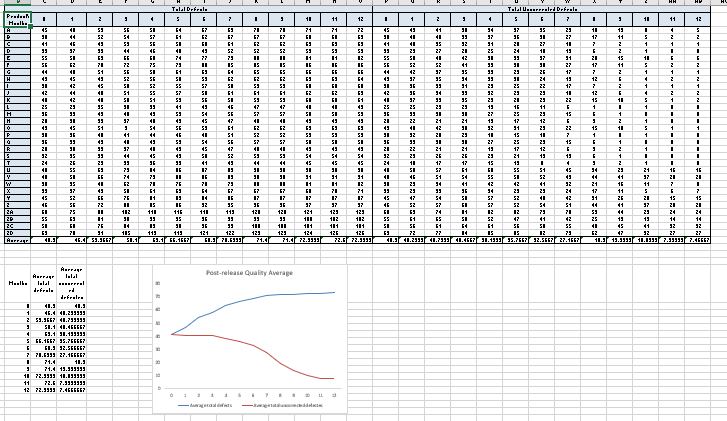
* **Worksheet2:** Post Release Quality of product ZD

Total defects and total uncorrected defects of product ZD shown in this worksheet, which is derived from the raw data in worksheet 1.



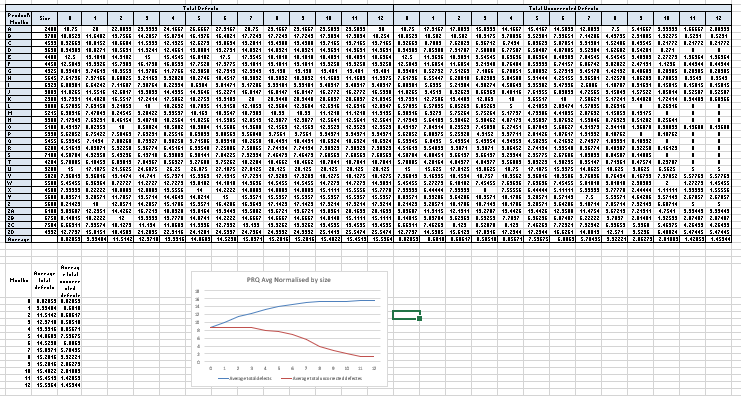
* **Worksheet3:** Post Release Quality Average

This worksheet shows the average total defects and average total uncorrected defects of all 30 products individually.



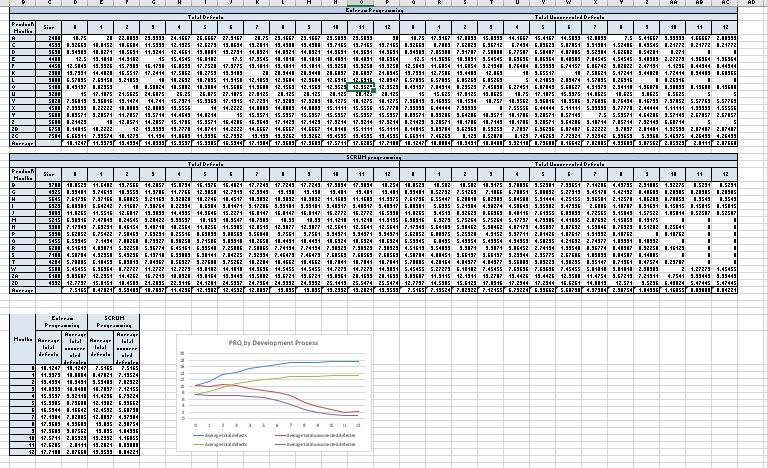
* **Worksheet4:** Post Release Quality Average Normalize by size

In the fourth worksheet PRQ Avg Normalized by Size, I have used the data from the third worksheet and normalize it by size by dividing each product’s defect count by its size and then multiplying it with 1000.



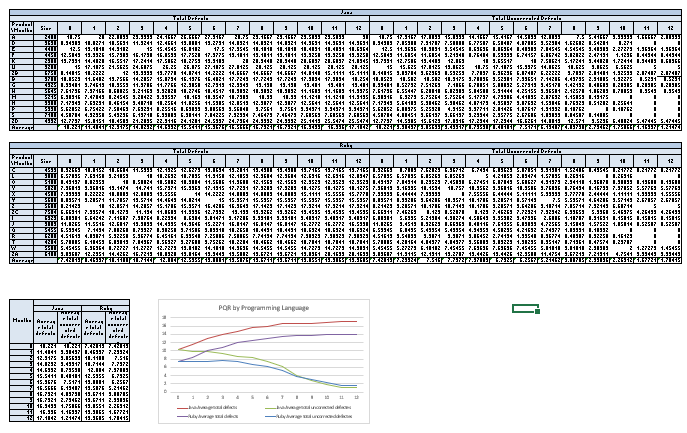
* **Worksheet5:** Post Release Quality by development process

I have used normalized data from worksheet 4. The data has been categorized in two type of development process Extreme Programming and Scrum by referring worksheet 1. Then average has been calculated for all 30 products.



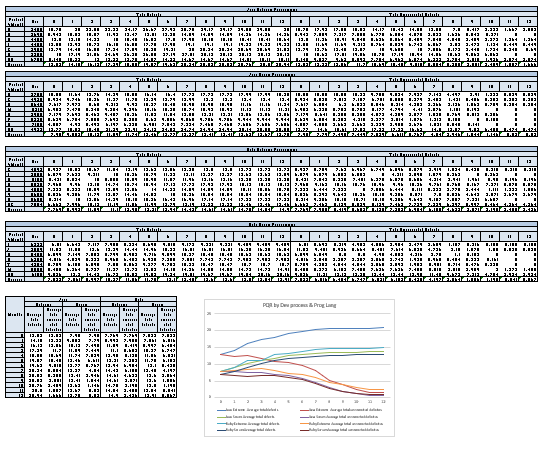
* **Worksheet6:** Post Release Quality by Programming Language

I have used normalized data from worksheet 4. The data has been categorized in two type of programming language Java and Ruby by referring worksheet 1. Then average has been calculated for all 30 products.



* **Worksheet7:** Post Release Quality by Development Process and Programming Language

I have used normalized data from worksheet 4. The data has been categorized in two type of development process Extreme Programming and Scrum by referring worksheet 1. And further categorized in two type of programming language used Java and Ruby by referring worksheet 1. Then average has been calculated for all 30 products.



1. **Measures, Graphs and Analysis**
   1. **Post Release Quality**
      1. **Overview**

Post Release Quality is measured for an individual product or for any collection of products (such as all those written in Java) and is defined as the number of defects in the product or collection after release. I have analyzed given 3 years of data for 30 products. The goal is to find how many known defects are there in the product and how does the total grow after release.

All the generated graphs are **line charts** and all the graphs uses following information:

* **Total Defects** at month T is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects)
* **Uncorrected Defects** is defined as Total Defects minus the cumulative sum of all defects that have been corrected.
  + 1. **Post Release Quality for Product ZD**

I have used data from worksheet 1 to calculate total defects and total uncorrected defects for the product ZD and generated a line graph for that.

**2.1.2.1 Graph**

The graph below shows Post Release Quality for one year of Product ZD.

X- axis: Number of Months

Y- axis: Number of Defects

**2.1.2.2 Analysis and Discussion**

By analyzing above graph, total defects were increases since product launched. On the other side defect correction first decreased for six months than increased since total uncorrected graph increases for six months then decreases.

**2.1.2.3 Procedure Used to Collect and Refine Data and Produce Graph**

With the help of N(new defects) and C(corrected defects) values from worksheet 1, total defects and total uncorrected defects has been calculated. To calculate total defects for a specific month, add N for that month with N’s from all previous months. To calculate total uncorrected defects for a specific month, subtract the sum of corrected defects till that month with the total defects of that month.

|  |
| --- |
| 2.1.2.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff.   The above data are collected separately for each software product. |

|  |
| --- |
| 2.1.2.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   Equation - Uncorrected Defects |

|  |
| --- |
| 2.1.2.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| * For this graph, the total defects are computed each month by adding the latest month’s defects to the previous total. * The uncorrected defects are computed each month by subtracting the number of defects corrected in the latest month from the previous uncorrected defects total. |

|  |
| --- |
| 2.1.2.3.4 How to Interpret the Graph |
| * Two lines are shown: both DT and DUNC,T are plotted monthly on a line chart for the first release of product A. The horizontal axis is number of months since product release and the vertical axis shows defect count. The chart shows one year of data. * The DT line is monotonic – it will tend to grow slowly, but to flatten out over time, as the product’s defects are found. When the line flattens, it is approximately equal to the total number of defects in the product. * The DUNC,T line should decrease over time, as defects are found and corrected. However it may increase in any particular month if a lot of new defects are found in that month. |

* + 1. **Post Release Quality Average**

This line graph represents post release quality of all 30 products by average of new defects and average of uncorrected defects by each month.

**2.1.3.1 Graph**

The graph below shows Post Release Quality Average for all 30 products.

X-axis: Number of Months

Y-axis: Average number of Defects per Product

**2.1.3.2 Analysis and Discussion**

From the graph it can be noticed that since the release of the all products more number of defects are identified for first two quarter of the year then after it almost stabilizes which represents less number of defects in the product. Above conclusion supported by the graph of the average total uncorrected defects which decreases in last two quarters.

**2.1.3.3 Procedure Used to Collect and Refine Data and Produce Graph**

The data worksheet a4data contains two values: N (Number of new defects) and C (Number of defects corrected) for each month. To calculate total defects for a specific month, add N for that month with N’s from all previous months. To calculate total uncorrected defects for a specific month, subtract the sum of corrected defects till that month with the total defects of that month. All the values of total defects and uncorrected defects for each month of every product are taken into consideration and average is calculated. A line graph is produced by taking months since release on horizontal axis and average number of defects per product on the vertical axis.

|  |
| --- |
| 2.1.3.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff.   The above data are collected separately for each software product. |

|  |
| --- |
| 2.1.3.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   Equation - Uncorrected Defects   * **Average total defects** is the sum of total defects for all months divided by the number of months.   Average case value   * **Average total uncorrected defects** is the sum of total uncorrected defects for all months divided by the number of months.   Average case value |

|  |
| --- |
| 2.1.3.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| For this graph, the total and uncorrected defects for all products, every month is computed. Then, the average for total and uncorrected defects is computed for each product. We copy the average total defects and average total uncorrected defects and place them in a table by their respective months. We plot graph using this table. |

|  |
| --- |
| 2.1.3.3.4 How to Interpret the Graph |
| * Two lines are shown: both average total defects and average total uncorrected defects are plotted monthly on a line chart for the Post Release Quality Average. The horizontal axis shows number of months since product release and the vertical axis shows the defect count. The chart shows one year of data. * Reading the total defects line: it grows slowly, but flattens out over time, as the months pass by. * Reading the total uncorrected defects line: uncorrected defects decrease with the months, as the errors are corrected with time. |

* + 1. **Post Release Quality Average Normalized by Size**

This graph shows the post release quality average of all the 30 products normalize by size of the code. For that each product’s defect count is divided by the product size and multiplied by 1000 to produce defects per thousand line of code.

**2.1.4.1 Graph**

The graph below shows Post Release Quality Average Normalized by Size for all 30 products.

X-axis: Number of Month

Y-axis: Defects per 1000 line of code

**2.1.4.2 Analysis and Discussion**

The defects per KLOC (1000 lines of code) value is calculated because it acts as a standard value/unit which can be used as a reference to compare the values between the 30 products over 3 years of time. By calculating the defects per KLOC the efficiency of an approach can be easily understood.

**2.1.4.3 Procedure Used to Collect and Refine Data and Produce Graph**

The data worksheet 1 contains two values: N (Number of new defects) and C (Number of defects corrected) for each month. To calculate total defects for a specific month, add N for that month with N’s from all previous months. To calculate total uncorrected defects for a specific month, subtract the sum of corrected defects till that month with the total defects of that month. Total number of uncorrected defects and total defects are divided by total size of products and then multiplied by 1000. Compute the average of the normalized values for total defects and total uncorrected defects. A line graph is produced for the two averages by taking months since release on horizontal axis and Defects per 1000 lines of code on the vertical axis.

|  |
| --- |
| 2.1.4.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff.   The above data are collected separately for each software product. |

|  |
| --- |
| 2.1.4.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   For each product, Defects/KLOC is calculated for all the months by dividing the total defects/total uncorrected defects by the size of the product and multiplying by 1000. We calculate the average of Defects/KLOC for each month. The computed values of average total defects/KLOC and average total uncorrected defects/KLOC is copied into a new table and a graph is plotted with the values from the two columns. Horizontal axis depicts months since release and vertical axis depicts Defects/KLOC.   * **SR -** Size of the product for release R * **Defects/KLOC-** Divide each product’s defect counts by the product’s size (and then multiply by 1000)   The above data are collected separately for each release of each product.   * **Average Defects/KLOC-** Divide sum of Defects/KLOC for a month by total number of products which are active for that month.   *Average defects/KLOC/total number of active products* |

|  |
| --- |
| 2.1.4.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| For each product, Defects/KLOC is calculated for all the months by dividing the total defects/total uncorrected defects by the size of the product and multiplying by 1000. We calculate the average of Defects/KLOC for each month. The computed values of average total defects/KLOC and average total uncorrected defects/KLOC is copied into a new table and a graph is plotted with the values from the two columns. Horizontal axis depicts months since release and vertical axis depicts Defects/KLOC. |

|  |
| --- |
| 2.1.4.3.4 How to Interpret the Graph |
| * Two lines are shown in the graph: both average total defects per KLOC and average total uncorrected defects per KLOC are plotted monthly on a line chart for the post release quality average normalized by size. The horizontal axis represents the months since release and the vertical axis shows number of defects/KLOC. The chart shows one year of data. * The total defects line is monotonic i.e. it grows slowly, but flattens out over time, as the months pass by.   The uncorrected defects line decreases over time, as defects are found and corrected with months. |

* + 1. **Post Release Quality by Development Process**

This graph represents relation between development process and defects for all the 30 products.

**2.1.5.1 Graph**

The graph below shows Post Release Quality by Development Process for each of the two processes used: Extreme Programming and SCRUM.

X-axis: Number of Months

Y-axis: Defects per KLOC

**2.1.5.2 Analysis and Discussion**

The above line chart has 4 lines, two for each development process named Scrum and Extreme Programming. It can be observed from the chart that the total defects started at around10 defects per KLOC during the release month for extreme programming, whereas, around 7.5 defects per KLOC for Scrum. Overall trend is almost similar for both the development processes as the number of months increased the defects increased at the same rate. For the total uncorrected defects, both the processes followed a common rate of decrease in the total uncorrected defects. Hence, looking at the graph it can be concluded that Scrum is a slightly better development process than extreme programming because of the lesser number of errors.

**2.1.5.3 Procedure Used to Collect and Refine Data and Produce Graph**

|  |
| --- |
| 2.1.5.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff. |

|  |
| --- |
| 2.1.5.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   Equation - Uncorrected Defects   * **SR -** Size of the product for release R * **Defects per KLOC** is defined as the number of defects for every thousand lines of code. It is calculated by dividing the number of defects with the product size and multiplying the result with 1000.      * **Average defects per KLOC** is defined as the average of defects per KLOC calculated for all the 30 products every month.   *Average defects/KLOC/total number of active products* |

|  |
| --- |
| 2.1.5.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| After calculating the average defects per KLOC for all the 30 products for each month, these averages are extracted and exported to a new table which has 5 columns: Months, Total defects per KLOC for XP, Total Uncorrected defects per KLOC for XP, Total defects per KLOC for Scrum, Total Uncorrected defects per KLOC for Scrum  The new table is used to plot the graph because it has all the elements in an organized fashion. |

|  |
| --- |
| 2.1.5.3.4 How to Interpret the Graph |
| The 4 lines on the graph represent Total defects per KLOC for XP(Red), Total uncorrected defects per KLOC for XP(Green), Total defects per KLOC for Scrum(Purple) and Total uncorrected defects per KLOC for Scrum(Blue).  The horizontal axis represents the months since the product’s release, whereas, the vertical axis represents the defects per KLOC. |

* + 1. **Post Release Quality by Programming Language**

This graph represents relation between programming language and defects for all the 30 products.

**2.1.6.1 Graph**

The graph below shows Post Release Quality by programming language for each of the two languages used: Java and Ruby.

X-axis: Number of Months

Y-axis: Defects per KLOC

**2.1.6.2 Analysis and Discussion**

The above line chart has 4 lines, two (total defects, total uncorrected defects) for each programming language. As it can be observed from the chart, the total defects started at around 10.2 defects per KLOC during the release month for Java, whereas, around 7.4 defects per KLOC for Ruby. Overall trend is almost similar both the programming languages as the number of months increased the defects increased at the same rate and almost flatten out at the 7th month. For the total uncorrected defects, the line representing Java dropped below Ruby, suggesting Java has corrected more defects than Ruby. And, overall, total defects started at a higher number for Java than Ruby and ended at an even higher number, which shows that even if the defects are higher when released in Java, but due to the impressive defect correction Java ended up having less number of errors. Hence, it can be concluded that Java is a better programming language than Ruby.

**2.1.6.3 Procedure Used to Collect and Refine Data and Produce Graph**

|  |
| --- |
| 2.1.6.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff. |

|  |
| --- |
| 2.1.6.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   Equation - Uncorrected Defects   * **Defects per KLOC** is defined as the number of defects for every thousand lines of code. It is calculated by dividing the number of defects with the product size and multiplying the result with 1000.      * **Average defects per KLOC** is defined as the average of defects per KLOC calculated for all the 30 products every month.   *Average defects/KLOC/total number of active products* |

|  |
| --- |
| 2.1.6.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| After calculating the average defects per KLOC for all the 30 products for each month, these averages are extracted and exported to a new table which has 5 columns: Months, Total defects per KLOC for XP, Total Uncorrected defects per KLOC for XP, Total defects per KLOC for Scrum, Total Uncorrected defects per KLOC for Scrum  The new table is used to plot the graph because it has all the elements in an organized fashion. |

|  |
| --- |
| 2.1.6.3.4 How to Interpret the Graph |
| The 4 lines on the graph represent Total defects per KLOC for Java(Red), Total uncorrected defects per KLOC for Java(Green), Total defects per KLOC for Ruby(Purple) and Total uncorrected defects per KLOC for Ruby(Blue).  The horizontal axis represents the months since the product’s release, whereas, the vertical axis represents the defects per KLOC. |

* + 1. **Post Release Quality by Development Process and Programming Language**

This graph gives us more detailed relation among programming languages and development process. Combining all together I got 8 different combinations.

**2.1.7.1 Graph**

The graph below shows Post Release Quality by two development processes and two programming languages: Extreme Programming, Scrum and Java, Ruby.

X-axis: Number of Months

Y-axis: Defects per KLOC

**2.1.7.2 Analysis and Discussion**

The above line chart has 8 lines representing all the combinations of programming language, development process and type of defect (total / total uncorrected). As it can be observed in the chart, the total defects started highest at 10.2 defects per KLOC during the release month for products developed in Java following extreme programming process, whereas, the rest of the combinations start almost at the same number in the range of 7.0 to 7.8, with Ruby in Scrum being the least. For the total uncorrected defects, all the combinations of programming languages and development processes showed good progress in reduction in the numbers. But, products developed in Java using Extreme Programming showed a drastic decrease in the number of total uncorrected defects. Overall, it is the products developed in Java and Scrum development process combination got impressive results. Hence, it can be concluded that the products developed in Java using Scrum development process are better than the others.

**2.1.7.3 Procedure Used to Collect and Refine Data and Produce Graph**

|  |
| --- |
| 2.1.7.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff. |

|  |
| --- |
| 2.1.7.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation 1 - Total Defects   * **Uncorrected Defects** (at month T is defined as **Total Defects** minus the cumulative sum of all defects that have been corrected:   Equation 2 - Uncorrected Defects   * **Defects per KLOC** is defined as the number of defects for every thousand lines of code. It is calculated by dividing the number of defects with the product size and multiplying the result with 1000.      * **Average defects per KLOC** is defined as the average of defects per KLOC calculated for all the 30 products every month.   *Average defects/KLOC/total number of active products* |

|  |
| --- |
| 2.1.7.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| After calculating the average defects per KLOC for all the 30 products for each month, these averages are extracted and exported to a new table which has 9 columns: Months, Total defects per KLOC for Java in XP, Total Uncorrected defects per KLOC for Java in XP, Total defects per KLOC for Java in Scrum, Total Uncorrected defects per KLOC for Java in Scrum, Total defects per KLOC for Ruby in XP, Total Uncorrected defects per KLOC for Ruby in XP, Total defects per KLOC for Ruby in Scrum, Total Uncorrected defects per KLOC for Ruby in Scrum.  The new table is used to plot the graph because it has all the elements in an organized fashion. |

|  |
| --- |
| 2.1.7.3.4 How to Interpret the Graph |
| The 8 lines on the graph represent Total defects per KLOC for Java in XP (Blue), Total uncorrected defects per KLOC for Java in XP(Red), Total defects per KLOC for Java in Scrum (Green), Total uncorrected defects per KLOC for Java in Scrum(Purple), Total defects per KLOC for Ruby in XP(Light Blue), Total uncorrected defects per KLOC for Ruby in XP(Orange), Total defects per KLOC for Ruby in Scrum(Dark Blue) and Total uncorrected defects per KLOC for Ruby in Scrum(Brown).  The horizontal axis represents the months since the product’s release, whereas, the vertical axis represents the defects per KLOC. |

* 1. **Post Release Quality History**
     1. **Overview**

Post release quality history is measured for a collection of products. It shows whether there is any relationship between the level of quality and when the product was released. This is a “lagging” indicator, which means it can only be computed after a release has been out for 12 months. The measure indicates the total number of defects found for all products in the collection during their first 12 months of use. It can be calculate quarterly or yearly.

* + 1. **Post Release Quality History By Quarter**

Data from worksheet 1 has been analyzed by quarter (3 months) manner and graph has been generated for all the products.

**2.2.2.1 Graph**

The graph below shows Post Release Quality History by Quarter

X-axis: Quarter

Y-axis: Total number of defects

**2.2.2.2 Analysis and Discussion**

There are 3 lines in the graph which represent the Worst case, Best case and Average of total number of defects that were recoded with the products released during that quarter. The graph started at a moderate 63, 68 and 72 for Best, Average and Worst cases respectively. After that it can be noticed steady increase in the worst-case number till Q3. In Q4, least numbers are can be seen in all three cases since Q1. After the Q4, the trend of constant reduce in the numbers can be observed till Q8. Hence, by examining this graph and other graphs shown in 2.1.5.1, 2.1.6.1 and 2.1.7.1, it is proved that scrum methodology allows less number of defects when compared to Extreme Programming.

**2.2.2.3 Procedure Used to Collect and Refine Data and Produce Graph**

|  |
| --- |
| 2.2.2.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff. * **Q** – Quarter of the year |

|  |
| --- |
| 2.2.2.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation 1 - Total Defects   * Best Case: Find the product with least in the quarter. * Worst Case: Find the product with highest in the quarter. * Average: Average = |

|  |
| --- |
| 2.2.2.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| After finding and calculating the Best, worst and average cases for each quarter, export the respective values into a new 4-column table which has Quarter, Best case, Average and Worst case as columns. Using the new table, a graph is plotted by quarter, using a line chart which has quarters on the horizontal axis and number of defects on the vertical axis. |

|  |
| --- |
| 2.2.2.3.4 How to Interpret the Graph |
| There are 3 lines in the graph: Orange represents Worst case, Blue represents average and Green represents Best case. The vertical axis represents the total number of defects and horizontal axis shows the quarters. It can be noticed that the best-case line is always below the average and worst-case lines, proving the authenticity of the measure. |

* + 1. **Post Release Quality History by Year**

Data from worksheet 1 has been analyzed by yearly manner and graph has been generated for all the products.

**2.2.3.1 Graph**

The graph below shows Post Release Quality History by Year

X-axis: year

Y-axis: total number of defects

**2.2.3.2 Analysis and Discussion**

There are 3 lines in the graph which represent the Worst case, Best case and Average of total number of defects that were recoded with the products released during that year. The graph started at a moderate 53, 67.8 and 83, for Best, Average and Worst cases respectively. After the year 2015, a drop in the number of the defects can be noticed in the year 2016. After 2016, there has been a significant increase in the number of defects because almost all the products were made using the Extreme Programming. Hence, once again, by this measure, it is proved that scrum methodology allows less number of defects when compared to Extreme Programming. But, with this measure it can’t be judged or analyzed for sure that because of the development process or programming language are affecting the number of defects. This is because, the company has released many products in the span of a year which have multiple combinations of programming languages and development processes. Hence, the previous measure, i.e., Post Release Quality History by Quarter is efficient and accurate in analyzing the data.

**2.2.3.3 Procedure Used to Collect and Refine Data and Produce Graph**

|  |
| --- |
| 2.2.3.3.1 Data Collection: Base Metrics Collected |
| The data required for this measure are:   * **DPRE** – The number of known defects at the time of product release. Collected at time of product release. * **DRPT, i** – The number of defects reported in the ***ith*** customer failure report. Collected at the beginning of each month. * **N** – The total number of customer failure reports (total number of months). This is normally 12 for each product. * **DC,i** – The number of defects corrected in month ***i***, reported monthly by engineering staff. * **Y** – Year |

|  |
| --- |
| 2.2.3.3.2 Compound Metrics (Metrics Computed) |
| * **Total Defects** at month T (is defined as the cumulative sum of all defects known at month T. This measure is computed every month, for each product. This includes both defects reported by the customer (post-release defects) and defects known to be in the product at release time (pre-release defects):   Equation 1 - Total Defects   * Best Case: Find the product with least in the year. * Worst Case: Find the product with highest in the year. * Average: Average = |

|  |
| --- |
| 2.2.3.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| After finding and calculating the Best, worst and average cases for each year, export the respective values into a new 4-column table which has Year, Best case, Average and Worst case as columns. Using the new table, a graph is plotted by quarter, using a line chart which has quarters on the horizontal axis and number of defects on the vertical axis. |

|  |
| --- |
| 2.2.3.3.4 How to Interpret the Graph |
| There are 3 lines in the graph: Orange represents Worst case, Blue represents average and Green represents Best case. The vertical axis represents the total number of defects and horizontal axis shows the quarters. It can be noticed that the best-case line is always below the average and worst-case lines, proving the authenticity of the measure. |