STATEMENT OF WORK

2

Assignment 2

CSE 6329 – 2019 Fall

Data Analysis Report based on Defect Reporting and Analysis Tool

**Summary**

Your company frequently develops software and has been receiving complaints from customers about high failure rates in some of your software. The maintenance and support staff have concluded that most failures are due to defects in the software that were not known at the time the software was released (in other words, your testing did not find these defects).

The CEO of the company has demanded that the software development team analyze the situation and recommend what to do. The software team has been collecting 12 months of post-release defect data for each software product for three years (2015-2017). However, due to a cost reduction program initiated by the CEO in 2018, they have stopped collecting post-release data for all new software projects, starting in 2018. They continued collecting “first 12 months” of post-release data for 2017 projects.

Here are some recent company actions that may be related to this problem:

* As mentioned above, the software team stopped collecting post-release defect data for new products in 2018. So the data spreadsheet contains data only for products released during a three year period – 2015 through 2017.
* A rudimentary quality assurance program existed in 2015 but was abandoned in 2016 due to a cost reduction initiative. A more comprehensive quality assurance program was initiated in in 2017 but abandoned in 2018, as part of that year’s cost reduction program. In 2019, neither the quality assurance program nor the data collection program has been reinstated.
* After having programmed exclusively in C++ since 2000, the company started using Java for about half of the projects starting in 2010.
* The company also started using Extreme Programming in 2010 as their primary development process, but added SCRUM as an alternative process in 2015. Since then, about half the projects use SCRUM and the other half use Extreme Programing.

The software team leaders recently had a meeting to discuss this situation. They had not analyzed any of the data, but based on their personal opinions they came up with the following possible explanations (hypotheses) for the customer complaints:

1. **There are not any significant failure rate increases**. “Customers always complain.” This view was expressed by some of the software developers, who don’t regularly get involved with customers.
2. Increases in defects are due to **programming language**. (The programmers hotly debate whether Java or C++ generates more defects).
3. Increases in defects are due to **software development process**. I.e., using SCRUM instead of Extreme Programming. (Again, the programmers disagree regarding which development process generates more defects).
4. Increases in defects are probably due to the **size** of the software. Our **newer products are bigger** and thus have more lines of code. The defect rate per line of code probably hasn’t changed.
5. Increases in defects are due to the fact that we have **more products in use**. If we normalized by the number of products, we would probably see no significant differences.
6. When the company instituted the comprehensive **quality assurance (QA) program** in 2017, some people believe that things started to improve, but others thought there was no significant difference. Those who did not believe the QA program improved things convinced management to abandon the program in 2018. But others believe that when they abandoned this program in 2018, defects started to get more numerous. (The supporters of the QA program also believe that the 2015 QA program was helping reduce defects, despite the fact that they were just learning how to do quality assurance back in 2015.)

After the meeting, the software team asked you to analyze the data and to report on what conclusions you can draw from this about the defect rate issue. Your goal is to understand the relationship, if any, between post-release defects (defects found after release to the customer) and the possible explanations/hypotheses listed above.

To do this, you examine the examples shown in the lectures and you select the following measures to help analyze the situation:

|  |  |  |
| --- | --- | --- |
| **Measure #** | | **Description** |
| 1 | | Post-Release Quality (line chart) **(4 graphs - see note 1, below)** |
|  | 1a | Post-release Quality Average for all products, normalized by size of software product |
|  | 1b | Post-release Quality Average for all products, by development process |
|  | 1c | Post-release Quality Average for all products, by programming language |
|  | 1d | Post-release Quality History |
| 2 | | Current Quality (line chart) **(3 graphs - see note 2, below)** |
|  | 2a | Current Quality Total |
|  | 2b | Current Quality Total, normalized by ***size*** of currently active product releases (defects per KLOC) |
|  | 2c | Current Quality Total, normalized by ***number*** of currently active products (defects per active product) |

Each of these measures and corresponding graphs requires that you refine and analyze the data in a different manner. To do this analysis, you use a Microsoft Excel workbook called DRAT (Defect Reporting and Analysis Tool). This workbook has a separate worksheet for each of the graphs. Each of the various worksheets may be separate tabs within the same Excel workbook. (This will make it easier to copy data from one worksheet to another.)

Data collected are available in the form of a spreadsheet. (You are given data for 30 products that have been released to customers over the three years 2015-2017.) Your task is to analyze and graph the data so as to provide useful information about the defect rate. Your deliverable (section 4.0) is a **report** that explains the analysis process, the measures and the graphs, and your conclusions and recommendations. This report will be presented to others in your company, some of whom have little knowledge of software or metrics, but who have a lot of influence. Note that, since your company has not done analysis of their measured defect data in the past, your report must explain what is going on in addition to explaining the results of your analysis. For this reason, the report requires certain explanatory information in addition to your analysis results. The report is in the form of a Microsoft Word® document.

**Background Information -- Typical Software Projects**

The typical software project in your company produces a new product release every year or so. Since there are ten projects, that averages to slightly less than 1 release per month or about 10 product releases per year. The previous release of the product is typically phased out shortly after the next release becomes available. Thus, after the startup period in 2015, there are normally about 10 active products in use at any time.

At the time of product release, you typically know about existing problems in the software and can document the number of known defects. After release, customers inform the company of additional problems. Your organization has reached agreement on what constitutes a "defect".

**Notes on the Measures**

**NOTE 1:** ***Post release quality*** answers the question "how many known defects are in this product (or group of products) and how does the total grow after release?" It can be 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 of products} each month after release (for 12 months after release). This is recorded each month for twelve months after a product is released and is displayed using two lines: ***total defects*** and ***total uncorrected defects***. A value for each line is calculated each month for the 12-month period after product release. This graph is illustrated in the class lectures (also see below). For the present assignment, you should produce four post-release quality graphs:

1. **Post-release Quality Average Normalized by Size** (section 2.1.a of your report). This graph should have a horizontal axis showing *months since product release* (0 through 12), and should show the average number of defects for all 30 products, **normalized by size (lines of code)**. The vertical axis will be *defects per 1000 lines of code*.

**Data Refinement.** To determine the values to be plotted on the graph, you must first shift the data so that, instead of being organized by calendar month, it is organized by months since product release. Then you must compute each product’s total defects and total uncorrected defects each month. Finally, you must divide each product’s defect counts each month by the product’s size (and then multiply by 1000) to produce “defects per 1000 lines of code”. Then you compute the average for each month for all products. Note that “month” in the horizontal axis means “month since product release”.

In this case, there will be two lines

* + total defects (average, normalized),
  + total uncorrected defects (average, normalized),
* Your report should discuss the reasons for normalization.

1. **Post-release Quality by Development Process** (normalized by size)(section 2.1.b of your report). This graph should be similar to the previous one except that you will show two lines for SCRUM projects and two lines for Extreme Programming projects.

* In this case, your report will discuss what differences there are and how significant they are (for example, whether one process produces more or fewer defects or uncorrected defects than the other.

1. **Post-release Quality by Programming Language** (normalized by size)(section 2.1.c of your report). This graph should be similar to the previous one except that you divide the products into those developed using Java and those developed using C++.

* In this case, your report will discuss what differences there are and how significant they are (for example, whether one language produces more or fewer defects or uncorrected defects than the other).

1. **Post-release Quality History** (not normalized) (section 2.1.d of your report). This measure takes a longer term look at post-release quality. It answers two questions:

* "overall, how good were the products we shipped each month, over several years, in terms of defect rates?”
* “how have things changed over time?"

This is measured for all products that were released during a particular time period. It is a “lagging” indicator that can only be computed after a release has been out for 12 months. The measure indicates the total number of defects found for all product releases during their first 12 months of use.

Since multiple product releases are shown, there are three values computed:

* best case defect count,
* worst case defect count, and
* average case defect count

for all products released during a given time period. This can be demonstrated using various charts (as illustrated in the class lectures).

For this assignment, use a ***line chart*** (3 lines: best, worst, and average). Each horizontal axis point represents a period of time (in this case, a quarter of a year, so there will be a total of 12 data points on the horizontal axis). The data represented for each point represents all products released during that time period (i.e., that quarter). The vertical value is the post release quality for all products released during that quarter (best, worst, and average). I.e., the total number of defects found after 12 months for all products released during that quarter.

The figure below illustrates a “by year” graph covering ten years (for this assignment you only cover three years but will graph the data by quarter).

To summarize, you should display **3 full years** of data, by quarter (each quarter represents anywhere from two to four product releases). Note that since the original data are provided by month, you will need to combine all data for a given quarter in order to graph by quarter.

# Example – Post Release Quality History

**NOTE 2:** ***Current quality*** is designed to answer the following question:

* "how many known defects are out there this month in all currently-active products?"

# Example – Current Quality

Current quality gives you a sense of how your customers see your products. If there are a lot of defects in your products, customers may not like this and may be seeking another company to buy their software from. Current quality information is a way to measure customer satisfaction, and it may also be helpful for deciding whether you need to assign more resources to testing or to defect correction. It is measured for all active products and is defined as the total uncorrected defects for all active products (all products that have been released and are still active – i.e., are still within their first 12 months). This is depicted as a line chart, as illustrated in the lectures and in the sample above. For each of the current quality totals (measures #2a, 2b, and 2c), graph 3 years of data by month. (The example above graphs 1 year of data by month.)

1. **Current Quality Total** (section 2.2.a of your report). Graph current quality, by month, for 3 years, using all active products each month. The vertical axis will be “total defects”.
2. **Current Quality Total Normalized by Size** (section 2.2.b of your report). The same, but each value is normalized by the size of the corresponding software product. In other words, for each month, normalize each active product’s defect level by size (as you did with post-release quality normalized by size) before computing the total. In this case, the vertical axis will be “total defects per thousand lines of code”.
3. **Current Quality Total Normalized by Number of Products** (section 2.2.c of your report). Similar to the above, but normalized by the total number of currently active products each month.

**In discussing graphs 2b and 2c, be sure to explain what each tells you that the previous ones do not, and whether that difference is important.**

**1.0 Work to be performed for this assignment**

**1.1 Recording and Analysis Tool.** You are to design and implement a defect recording and analysis tool (DRAT) using a workbook of spreadsheets (also known as worksheets). DRAT will be used for the following purposes:

1.1.1 ***Record*** the defect data (extract defect data from the data collection spreadsheet provided - or simply expand that spreadsheet - see section 4.0, below).

1.1.2 Help with ***refinement*** and ***analysis*** of the data (this will consist of such activities as shifting and sorting the data, extracting subsets of the data, and computing totals, averages and other measures from the data.) These actions may be performed manually or automatically if you wish, but don't get carried away with fancy spreadsheet tricks. [Hint: it will be easier if you use Excel formulas. Excel formulas are illustrated in several places in your A1 PWBS spreadsheet template. Studying these will help you understand how formulas work and how they can be used.]

1.1.3 Generate various ***graphs*** illustrating the measures. These graphs will assist in the analysis and communication processes.

**Special note: although most of the data analysis will be done using the DRAT tool, grading of this assignment will be based mainly on the quality of your report.**

**1.2 Documentation.** You will prepare a **Defect Analysis Report** (Appendix A and template), a Word® document that explains each measure and its graph, the measures and data that go into it, and how the data must be manipulated to calculate the measure and produce the graph, and what the graph tells you about the potential defect rates and correction rates. This report is to be presented to software developers and software managers, and you should assume they know how to use a spreadsheet program.

The report will contain the following:

**1.2.1 Analysis Tool Description.** (Section 1.0 of the defect analysis report.) A brief description of the format and structure of the spreadsheet you created, such as how you modified the initial data spreadsheet to facilitate data entry or perform your calculations. A few pages should be enough for this purpose.

**1.2.2 Individual Measure and Graph Descriptions.** (Section 2.0 of the defect analysis report.) There will be several pages of information about each measure and graph (section 2.1, 2.1.a, 2.1.b, 2.1.c, 2.1.d, 2.2, 2.2.a, 2.2.b, and 2.2.c), showing:

(2.n.m.i) an **overview** of the measure and graph. This section defines what is being measured and explains the information need and the question(s) being answered by the measure and its graph.

(2.n.m.ii) a **graph**, using data from your worksheets (see notes, above). In most cases you will produce the graph within the Excel workbook and copy it into the report;

(2.n.m.iii) **Analysis and Discussion**. You will explain what the graph tells you about the defect rates and the various hypotheses (see template for the format to be used). You will include a table showing what this measure/graph says about each of the five hypotheses. In the summary column of the table (see template) you will rate each hypothesis on the following scale:

-- The graph strongly refutes the hypothesis

- The graph refutes the hypothesis but not strongly

0 The graph says little or nothing about the hypothesis

+ The graph supports the hypothesis but not strongly

++ The graph supports the hypothesis strongly

**1.2.3 Summary and Recommendations.**  Section 3.0 of the report is where you take a look at all seven of the graphs and report on your overall assessment and recommendations.

Section 3.1 will assess each of the hypotheses against all seven of the graphs. For example, some hypotheses might be supported by multiple graphs, but others might have mixed results with some support and some refutations. You will use the following scale to give an overall assessment and then explain your reasoning:

-- The graphs strongly refute the hypothesis

For example there are multiple “-“ and “--" results and no “+” results.

- The graphs refute the hypothesis but not strongly

For example, there are several “-“ results but no “--" results and perhaps there’s also one “+”

1. The graphs say little or nothing or are inconclusive about the

hypothesis. For example, mostly “0” results or some “-“ and some “+” results.

+ The graphs support the hypothesis but not strongly

For example, there are several “+“ results but no “++" results and perhaps there’s also one “-”

++ The graphs supports the hypothesis strongly

For example there are multiple “+“ and “++" results and no “-” results

Section 3.2 will provide your overall recommendations. These are actions you recommend based on what you learned. For example, should they reinstate the quality assurance program? Switch to Java for all projects? (etc.)

A detailed outline of the report appears as Appendix A of this SOW. There is also a template for the report, which shows a suggested format and some further details.

**Note 3:** in order to produce your graphs, use your analysis tool (spreadsheet) with the data provided to you (see 4.0 – Data Spreadsheet).

**Hints**

Hint 1: I recommend that you use a workbook with multiple worksheets, (spreadsheets / tabs) to simplify moving data around. In many cases you can use a formula to copy data from one spreadsheet to another.

Hint 2: Sophistication of your analysis tool will not be a factor in your grade. *The grade is based on the quality of your report***,** reflecting the analysis done using the DRAT tool. However, you must turn in both the **DRAT tool** and the **report** so that if you make a mistake we can evaluate the tool to see what you did wrong. Grading will be based on how well you understand how to analyze the data and report your results. But learning how to utilize a spreadsheet for this purpose is a "bonus" learning experience from this assignment.

Hint 3: When copying graphs or spreadsheet excerpts into the report, I recommend you paste them as bitmaps. If you try to paste them as Excel objects, strange things may happen (such as messing up fonts).

**2.0 Defect Characteristics**

**2.1 Defect types and priorities.** There are many types of failures. For your type of software, it is generally straightforward to evaluate a failure and identify the defect causing it. This analysis is performed by the customer representative and the software maintenance organization, so by the time your organization sees the data you only see defects, not failures. The customer representative assigns a priority and a type to each defect, but you may ignore defect types and priorities for this assignment.

**2.2 Defect identification and collection prior to release.** The number of defects in each product at the time of release is the number of defects discovered prior to the release date that were not corrected by the time of release – i.e., defects found during development and testing that were not corrected. Although this assignment focuses on defects discovered after release, you may want to know how the organization determines defects prior to release. So here is how defects are detected prior to release.

Defect identification occurs at inspections, reviews, walkthroughs, and tests. Each time a defect is detected, it is counted and documented. Usually, most of these defects are corrected by the time a product is released – but not all of them – sometimes they are considered too minor to hold up product release. At the time of product release, the total number of defects that have not been corrected is shown as the “month 0” defect count.

**2.3 Defect identification and collection at and after release.** At release, all known defects that have not been corrected are counted and the total is documented in the spreadsheet provided to you.

After release, customer-reported problems are analyzed and when the defects are determined they are also recorded in the spreadsheet in the month where the problem was originally reported.

**3.0 Product Characteristics**

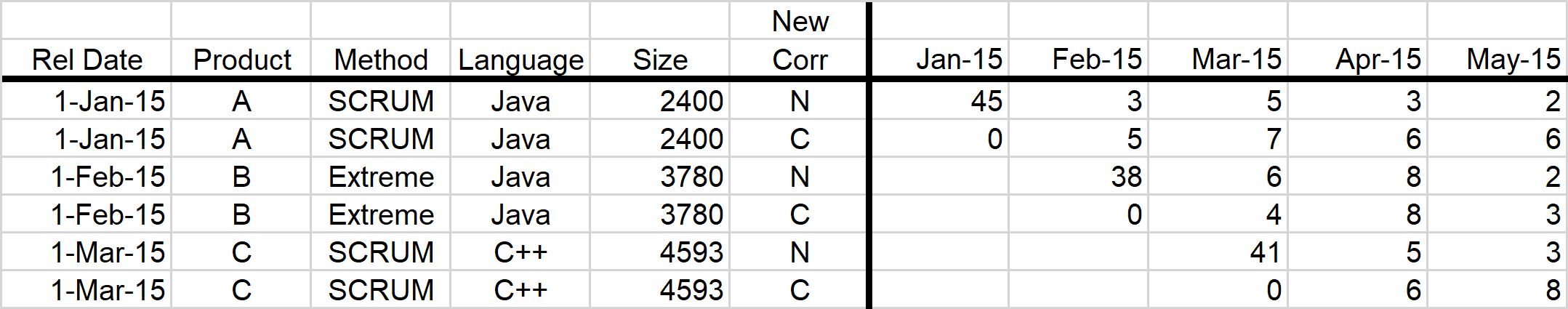
**3.1 Product identification.**  Each product has a unique identifier (A, B, C, …, Z, ZA, ZB, ZC, ZD). This information is recorded on the spreadsheet, along with the development process, programming language, release date and size in lines of code. This will make it possible to categorize and group products in various ways.

**3.2 Product lifetime.**  Each product release is generally removed from support approximately 12 months after it was released. This is because the product is replaced by the next release, which is (hopefully) a superior product. However, the data collection spreadsheet does not tell you which products are replacing which other ones, and that information does not matter to your report or analysis. Occasionally the spreadsheet may contain entries beyond the 12-month point, reflecting customer problems identified later. For purposes of this analysis, you should ignore data recorded after 12 months of product use (months 0-12).

**4.0 Data Spreadsheet**

The data provided consists of a Microsoft Excel spreadsheet named **A2DATA.XLSX**.

* The spreadsheet has two rows for each release of 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. The spreadsheet looks looks similar to the one shown below. (Note: assume size is in source lines of code). Note that the figure below shows only the first few rows and columns of the spreadsheet.



* **Rel Date** is the date when the product was released. **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.
* **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

Note that the data shown in the figure above are for illustration only. You should use the data in the spreadsheet provided to you, which may be different. Also note that the data shown are “beginning of month” figures and the very first month (which will be shown on graphs as month 0) is the number of defects in the product at the time it was released. For example, in the above figure, product A, release 1 was released in January 2015 and had 45 known defects. January 2015 is month 0 for this product. At the end of the first month, i.e., at the beginning of February 2015, 3 more defects had been discovered and 5 defects had been corrected. February 2015 is month 1. Product C was released in March 2015 and had 41 known defects. March, 2015 is month 0 for this product.

**5.0 Deliverables** You must deliver two files.(File names shown below are for individual assignments. If you are teaming, put both teammate names in the file name, as described in the module 00 course slides.)

* **DRAT Tool**. Your workbook (spreadsheet), as described above.
  + **Template:** None. You must create this yourself. I suggest starting with the data spreadsheet.
  + **File name of deliverable:**

**A2 CSE6329 2019fa DRAT last first.xlsx**

* **Defect Analysis Report.** See Appendix A for the outline and Appendix B for an example. Also see the report template for specific details.
  + **Template:**

**A2 CSE6329 2019fa - Report Template.docx**

* + **File name:**

**A2 CSE6329 2019fa - Report last first.docx**

* The Report has the outline shown in Appendix A and in the template.

***REMEMBER TO PUT YOUR NAME(S) ON THE FRONT OF YOUR REPORT BEFORE YOU TURN IT IN***.

# Appendix A

Detailed Outline of Defect Analysis Report.

Cover page (shows title of report and names of authors)

Grading Template page (student should not write on this slide)

Table of Contents

1. Introduction
   1. Purpose of This Report
   2. Structure of Analysis Tool
   3. Data Collection Overview
   4. Summary of Analyses and Graphs
2. Measures, Graphs and Analysis

2.1 Post-Release Quality

2.1.a First Graph/Metric

2.1.a.i Overview of graph/metric

* + - * Purpose / Information Need
      * Question being Answered by This Metric
      * Definition of the Metric
      * Collection Frequency
      * Type of graph (line chart, bar chart, etc.)

2.1.a.ii Graph

2.1.a.iii Analysis and Discussion

* + - * What the graph tells you about the defect rate and data correction rate

2.1.b Second Graph/Metric

(repeat as above for each of the remaining post-release quality metrics)

2.2 Current Quality

2.2.1 First current quality metric

(repeat as above)

…

3.0 Overall Recommendations

3.1 Summary

* + A table showing a summary of what each graph/measure tells you about each hypothesis.

3.2 Recommendations

* + Your recommendations about what to do to reduce defects and improve defect correction rates.

End of Presentation.