STATEMENT OF WORK

2

Assignment 2

CSE 6329 – 2018 Fall

Data Analysis Report based on Defect Reporting and Analysis Tool

**Summary**

Your company frequently develops software and has been collecting data for some time. However they haven’t done much with the data. The primary programming languages in use are Java and C++, and there are major debates among the programming staff as to which language produces higher quality results. There are also two development processes in use: Extreme Programming and SCRUM. As with languages, there are debates about which process achieves higher quality results.

The major goal of your organization is to produce high quality software, because customer complaints about software problems are resulting in loss of some customers. You have been provided with three years of defect data for your software products and have been asked to analyze the defect data and report on the results. The goal is to understand the relationship, if any, between defects found after release to the customer and the development process and language. You have also been asked to examine the level of defect correction, because part of the argument is that some languages/processes produce code that is harder to debug and correct.

You turn to 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) **(6 graphs - see note 1, below)** |
|  | 1a | Post-release Quality for product ZB |
|  | 1b | Post-release Quality Average for all products, normalized by size |
|  | 1c | Post-release Quality Average for all products, by development process |
|  | 1d | Post-release Quality Average for all products, by programming language |
|  | 1e | Post-release Quality History |
|  | 1f | Graph of Your Choice (does not have to be a line chart) |
| 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 requires that you refine and analyze the data in a different manner, so you use Microsoft Excel workbook, with 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 so far 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. Your deliverable (section 4.0) is a **report** that explains the analysis process, the measures and the graphs. 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 measurements 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 a lot of information in addition to your analysis results. The report is in the form of a Microsoft Word® document.

## Background Information

[*Note: you will need some of this information to find explanations for some of the behaviors observed in the graphs.*]

The organization decided to start collecting data in 2015 and the results showed they had more defects than they were comfortable with. In late 2015 they started to notice differences between the results of different projects, although there was not enough data to draw any definite conclusions about the reasons for these differences. By the end of 2017 the organization had three years’ worth of data for 30 products, so they want to see what can be learned from the data that were collected. The data cover products released during three years: 2015-2017. Data for 2018 show the post-release results for the products released in 2017. However no new releases are shown for 2018.

Another relevant factor is that in late 2015 your organization started a defect reduction program as part of a corporate “Total Quality” thrust. This program continued in 2016. However, in 2017 there was a major cost reduction effort that resulted in loss of funds for some of the quality initiatives. In 2018 the organization discontinued data collection for new product releases, although they continued to track existing releases.

**Additional Background Information -- Typical Software Projects**

The typical software project in your company produces a new product release every year or so. The previous release of the product is typically phased out shortly after the next release becomes available. Since there are ten projects, that averages to slightly less than 1 release per month or about 10 product releases per year. At the time of release, you typically know about existing problems in the software and can document the number of known defects. After release, customers inform you 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 and how does the total grow after release?" It 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 each month 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 six post-release quality graphs:

1. **Post-release Quality for One Product** (section 2.1.a of your report). The first graph should show ***one year*** of history for product **ZB**. **ZB** is one of the most recent of all the products shown in the data. The horizontal axis should be months since release, starting at 0 (the month of release) and continuing to month 12 (for a total of 13 months). The vertical axis should be total defects. The format of this graph is illustrated below (but you should create your own graph, using the data in your worksheet, and you may use line colors and thickness of your own choosing). **Note that the data spreadsheet may occasionally contain more than 12 months’ worth of data. Only the release month and the next 12 months should be used when producing the graph.**

* This graph will be used in your report to explain the basic facts about the post-release quality graph, such as what data refinement you must perform to produce the graph. See Appendix A and the template for the format of the report.

# Example – Post Release Quality

1. **Post-release Quality Average Normalized by Size** (section 2.1.b of your report). This graph should have the same horizontal axis as above (months since product release), but should show the average for all 30 products, **normalized by size**. The vertical axis will now be “defects per 1000 lines of code”. To determine the values to be plotted on the graph, you first divide each product’s defect counts 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.

In this case, there will be four lines

* + total defects (average, normalized),
  + total uncorrected defects (average, normalized),
  + total defects for product ZB (normalized),
  + total uncorrected defects for product ZB (normalized).
* Your report will discuss how product ZB compares with the average. (There is a lot of interest among your managers and software developers as to whether product ZB is any better than the average.)
* Your report will also discuss the reasons for normalization.

1. **Post-release Quality by Development Process** (normalized by size)(section 2.1.c of your report). This graph should be the similar to the previous one except that you do not need to show lines for product ZB. Instead 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.d 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.e of your report). This measure takes a longer term look at post-release quality. It answers the question, "overall, how good were the products we shipped over several years in terms of defect rates and 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 (such as a given month, quarter, or year) and the data represented for that point represents all products released during that time period. Thus if the horizontal axis is by 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.

A key question for any proposed metric is *how frequently to measure, monitor, analyze and display the metric*. Monitoring too often may cost a lot for little benefit; monitoring too little may fail to give insight into what is really happening. There has been considerable debate among the management and staff of your software organization as to how frequently the post release quality history should be analyzed – that is, whether the post release quality graph should be shown by month, by quarter or by year. Since in most months there is only one release (and none at all in some months), you have agreed that graphing by month is not of much value. But there are still strong opinions about whether to graph by quarter or by year. To help resolve this, you will show two graphs, one for each of these frequencies, and make observations as to how useful they are. In other words, for post-release quality history you will display 2 graphs: by quarter, and by year.

The figure below illustrates a “by year” graph covering ten years (for this assignment you only cover three years). You may find that only one of the two monitoring frequencies is useful, or you may find that each monitoring frequency gives a different perspective on the data, both of which are useful. You need to decide which frequencies are helpful and for what purpose.

To summarize, you should display **3 full years** of data in two ways: by quarter (in which case each quarter represents anywhere from two to four product releases), and by year, in which case each data point represents about ten 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, and combine all data for a given year in order to graph by year. Once you have produced two graphs, your discussions should compare the two graphs, explaining the advantages and drawbacks of each and when, if ever, each would be useful.

# Example – Post Release Quality History

1. **Graph of Your Choice** (section 2.1.f of your report). In this case you produce a sixth graph that shows something not apparent from the first five graphs. You need to think about what other concepts might be of interest, and examine the data to see if there is anything interesting enough to show. Points will be assigned for this graph based on how well your graph shows something worth knowing about the data.

* In this case, your report will discuss what your graph shows that is different from what the other graphs show and how you generated it (your data refinement, compound metrics used, etc.).
* Bonus opportunity: if your chosen graph uses some of the statistical functions mentioned during the lectures, you may receive bonus points (depending on the sophistication of the functions used and whether you used them correctly). Be sure to mention that you used such functions in describing how to generate your graph

**NOTE 2:** ***Current quality*** is designed to answer the following question: "how many known defects are out there this month in all currently-supported 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 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 why 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 sorting the data, extracting subsets of the data, 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 and assisting in the analysis and communication process.

**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, Appendix B 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. 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, 2.1.e, 2.1.f, 2.2, 2.2.a, 2.2.b, and 2.2.c), showing:

(2.n.m.i) an **overview** of the measure and graph(s). This section explains the information need and the question(s) being answered by the graph. If there are multiple graphs, explain briefly what each is for;

(2.n.m.ii) a **sample graph (or graphs)**, using data from your worksheets (see notes, above - in some cases more than one graph should be shown). 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:

(2.n.m.iii.1) **general discussion** of what this measure and graph tells you (how a manager or developer would use the graph, based on general characteristics of this type of graph, **not the specific data or the specific graph**). This section is intended to explain the graph to people who have never seen it before. Also feel free to discuss **pitfalls** to watch out for (ways the graph might be misinterpreted or misused). The discussion should also explain what this measure/graph tells you that the previous ones do not;

(2.n.m.iii.2) **specific discussion** of the sample graph(s) you produced (i.e., what that particular graph tells you, based on the specific data being shown, what corrective action or other action is recommended (if any), possible explanations for any noteworthy trends or effects in the graph, etc.). Note that it not sufficient to simply describe the graph, as in “the graph went up in 2017”. You must actually analyze what it means, as in “the graph went up in 2017, indicating an increase in the number of defects that year. This is possibly due to …” [description of why you think it might have gone up, based on the other information provided to you.]. Further analysis may also be appropriate (your job is to analyze, not just describe).

(2.n.m.iv) the **procedure** by which the data are refined and the graph is produced (how to manipulate the data in the worksheet, what data to use for what part of the graph, etc.). Also provide the name and definition of the measure being graphed, how many products are covered by the graph (one, several, all from one year, all products, etc.), and the time period covered.

A detailed outline of the report appears as Appendix A of this SOW. Appendix B of this SOW is a separate, Word® file, which gives an example of what a report might look like. There is also a template for the report, which shows a suggested format.

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

**Note 4:** in order to test that your procedure for producing the graph is correct (section 2.n.m.iv), follow that procedure when producing the sample graph(s).

**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 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, 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. However, if you wish, you may use one or both for the graph you choose yourself in the post-release quality section of the report. (If you do that you may need to make up some of the data. Be sure to explain that in your report.)

**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. Also, you may find this of interest for the graph you select yourself. 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**

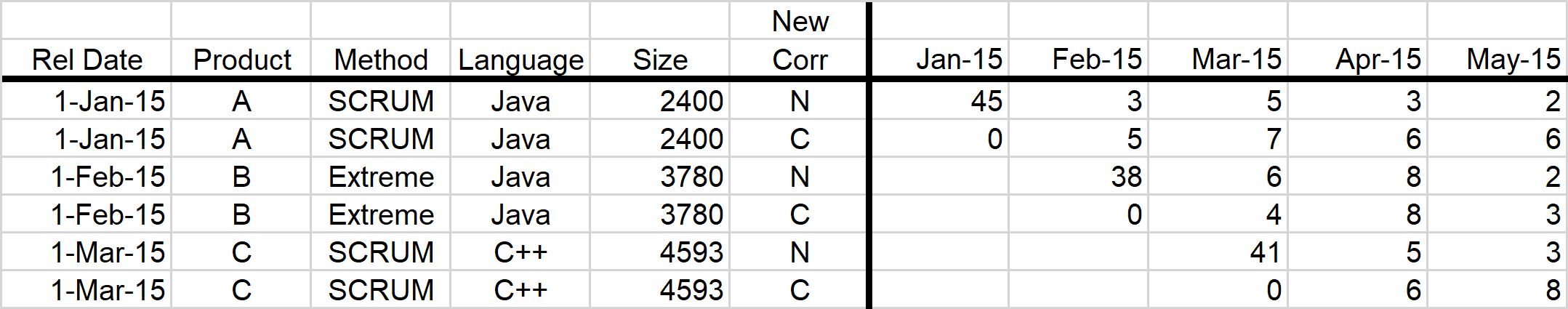
**3.1 Product identification.**  Each product has a unique identifier (A, B, C, …, Z, ZA, ZB, ZC, ZD). This information is recorded on the spreadsheet, along with the development process, programming language, release date and size in lines of code. This will make it possible to categorize and group 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 2018fa 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 2018fa – Report Template.pptx**

* + **File name:**

**A2 CSE6329 2018fa Report last first.pptx**

* 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 Sample Graphs

2.1.a.iii Analysis and Discussion

1 General Discussion

* Purpose
* Graph
* How to Analyze and Respond to the Graph

2 Specific Discussion

* + - * What the Graph Shows
      * Conclusions
      * Observations/Comments / Recommendations

2.1.a.iv Procedure for Collecting and Refining the Data and Producing the Graph(s)

* + - * Base Measures
      * Data Refinement
      * Compound Measures
      * Generating the Graph

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)

…

End of Presentation.

# Appendix B

Example of an Analysis Report

This is provided as a separate, Word® file.

Note that the example is for a different measure than the ones you will be reporting on, and is ***slightly different in format***. You should use the format in the template.