* STATEMENT OF WORK

2

Assignment 2

CSE 6329 – 2017 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 Ruby, 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 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.

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)** |
| 2 | Post-Release Quality History (line chart) **(2 graphs - see note 2, below)** |

Each of these measures requires that you analyze the data in a different manner, so you use a separate worksheet for each of the graphs. However 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 2014-2016.) 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 read by 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 2014 and the results showed they had more post-release defects than they were comfortable with. In late 2014 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: 2014-2016. Data for 2017 show the post-release results for the products released in 2016. However no new releases are shown for 2017.

**Notes on the Measures**

**NOTE 1:** ***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. It answers the question "how many known defects are in this product and how does the total grow 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*** 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 Product ZD** (section 2.1.2 of your report). The first graph should show ***one year*** of history for product ZD. ZD is 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 format of this graph is illustrated below (but you should create your own graph 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** (section 2.1.3 of your report). The second graph should show the ***average (mean) for all 30 products.*** The horizontal axis is the same - months since release. The vertical axis is average number of defects per product. As with the first graph, there will be two lines: total defects and total uncorrected defects.

* In this case, your report will discuss how product ZD compares with the average. (There is a lot of interest among your managers and software developers as to whether product ZD is any better than the average.)

1. **Post-release Quality Average Normalized by Size** (section 2.1.4 of your report). The third graph should show the average for all 30 products, as above, but this time the data for each product must be **normalized by size**. In other words, 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, as with the second graph. Again, there will be two lines – total and uncorrected. The vertical axis will now be “defects per 1000 lines of code”.

* In this case your report will discuss the reasons for normalization.

1. **Post-release Quality by Development Process** (section 2.1.5 of your report). The fourth graph should be the same as the third except that you divide the products into two categories: those developed using Extreme Programming and those developed using SCRUM. Thus you will have four lines – total defects per 1000 lines of code for Extreme Programming projects, total defects per 1000 lines of code for SCRUM projects, and total uncorrected defects for each development process.

* 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** (section 2.1.6 of your report). The fifth graph should be similar to the fourth except that you divide the products into two different categories: those developed using Java and those developed using Ruby. Thus you will have four lines – total defects per 1000 lines of code for Java projects, total defects per 1000 lines of code for Ruby projects, and total uncorrected defects for each programming language.

* 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. **Graph of Your Choice** (section 2.1.7 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. If you use 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.

* In this case, your report will discuss what your graph shows and how you generated it (your data refinement, compound metrics used, etc.).

**NOTE 2:** ***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 answers the question, "overall, how good were the products we shipped at different times in terms of defect rates?" Since multiple products are shown, there are three values computed: best case, worst case, and average case defect counts for all products in a given collection. This can be demonstrated using various charts (as illustrated in the 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 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 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 each of these frequencies and make suggestions as to how useful they are. In other words, for this assignment you will display post release quality history **two ways (2 graphs total)**: 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. 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

**Typical software projects**

Your company produces many software products and releases a new software product roughly once a month. {As you can see from the data, there are thirty products over a three year period, which averages to slightly less than 1 release per month.} 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".

**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, and so forth.) These actions may be performed manually or automatically if you wish. Don't get carried away with fancy spreadsheet tricks, but you should learn how to use Excel formulas and the Excel sort function, as this will greatly simplify your analysis and data refinement. [Excel formulas are illustrated in several places in your A1 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 and template), which explains each measure and its graph, the 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 shown to software developers and software managers, and you should assume they know how to use a spreadsheet program.

**Overview and 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. This is reported in section 1.0 of the defect analysis report.

**Individual Measure and Graph Descriptions.** (Section 2.0 of the defect analysis report.) See the template and Appendix A for further details.

Appendix A of this SOW gives an example of what a report might look like. There is also a template for the report, which shows the required format and provides suggestions about what goes into each section.

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

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

**Note 5**: the cover sheet for the report (contained in the template) shows how many points are assigned to each section of the report.

**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 changing fonts and font sizes.

**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 graph 6 in the post-release quality group.

**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 graph 6.) Defect identification occurs at inspections, reviews, walkthroughs, and tests. Each time a defect is detected, it is counted but not otherwise documented. Usually, most of these defects are corrected by the time a product is released – but not all of them. 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**

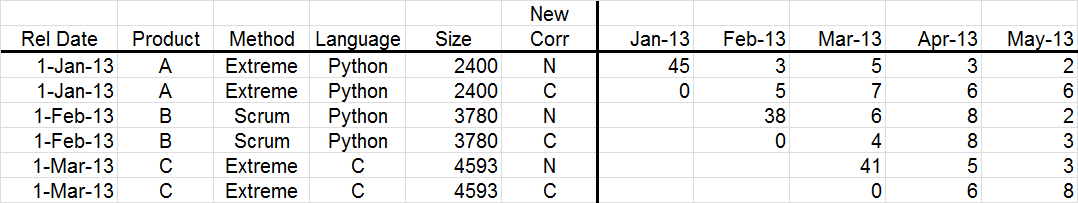
**3.1 Product identification.**  Each product has a unique identifier (A, B, C, …, Z, ZA, ZB, ZC, ZD). This information is recorded on the spreadsheet, along with the development process, programming language, release date and size in lines of code. This will make it possible to categorize and group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



* **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

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 was released in January 2013 and had 45 known defects. January 2013 is month 0 for this product. At the end of the first month, i.e., at the beginning of February 2013, 3 more defects had been discovered and 5 defects had been corrected. February 2013 is month 1. Product C was released in March 2013 and had 41 known defects. March, 2013 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 elsewhere.)

1. **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 2017fa DRAT last first.xlsx**

1. **Defect Analysis Report.** See Appendix A for an example. Also see the report template for more specific details.
   * **Template:**  **A02 CSE6329 2017fa - Report Template.docx**
   * **File name of deliverable:**

**A2 CSE6329 2017fa - Report last first.docx**

The Report format and outline are explained in the template and there is an example of what part of the report might look like in Appendix A.

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

# Appendix A

Excerpts from a sample Defect Analysis Report

**Defect Analysis Report**

**This example is for a different set of graphs from the ones you will be doing. Its only purpose is to give you an idea of what the report should look like.**

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

This report shows the results of analyzing why employees leave the company to work elsewhere (employee turnover). We have analyzed the data in six different ways. The goal is to determine whether turnover is a problem, whether it’s a problem for only certain projects or for the company as a whole, and what the likely causes are for undesired levels of turnover.

* 1. **Data Used**

The data necessary to perform these measurements and analyses have been collected monthly for each active software product, over the past three years. The data are stored in the **data spreadsheet**, named INPUTDATA. This spreadsheet has two sections. The first section, which shows actual turnover rates, has four rows for each project, then two rows for each software product developed by that project. Each month a new column is added and data collected for that month are recorded. The second section, which shows the results of exit interviews with departing employees, contains a list of reasons for employee dissatisfaction and the frequency with which each reason is cited.

A typical data collection worksheet looks like this:

*<Picture of worksheet goes here, showing typical situation.>*

* 1. **Analyses and Graphs**

We have analyzed the data in several different ways, resulting in six different metrics and their corresponding graphs. They break down into two categories: turnover data and data gathered from employee exit interviews. The six are:

1. Unplanned Turnover by Project,
2. Unplanned Turnover for the Whole Organization,
3. Turnover by year
4. Reasons most frequently cited for employees leaving
5. Reasons most frequently cited as things disliked about the company
6. Reasons most frequently cited as things liked about the company

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

* 1. **Structure of Analysis Tool**

In order to analyze turnover data, we have created an analysis tool consisting of a Microsoft Excel™ workbook that contains seven worksheets. The first worksheet contains the data collected each month (secton 1.2). The other six worksheets contain the data refinement and graphs corresponding to the six methods of analysis listed in section 1.3.

The other worksheets are: *<list each and describe each in an overview fashion, including a picture of each>.*

Details of each analysis are found in section 2 of this report.

Note that in this example, the six measures are divided into two groups. Section 2.1 describes the first group (turnover) and section 2.2 describes the second group (reasons cited for leaving, etc.)

1. **Measures, Graphs and Analysis**
   1. **Turnover (3 Measures)** 
      1. **Overview**

The turnover measures are used to determine correlation between turnover rates and other factors. *[Note that this sentence states the purpose of the measure or group of measures – the information need that applies to the entire group of measures.]* Three turnover measures are used: unplanned turnover for a project, unplanned turnover for the entire organization, and turnover by year. Unplanned turnover is defined as any turnover that exceeds planned turnover. *[This second sentence defines the measures in words.*] Turnover data are collected monthly starting at the beginning of each project. *[This sentence describes the frequency of data collection.]*

*< Additional descriptive information may also be supplied.>*

* + 1. **Unplanned Turnover for a Project**

Unplanned turnover is defined as any turnover that exceeds planned turnover. *[This sentence defines the measure in words.]* This measure is used to determine whether employee turnover is at a level that is likely to result in cost or schedule problems. *[Note that this sentence states the specific purpose of the measure – the information need.]* In the first measure, this is measured for a specific project.

Unplanned turnover helps understand the stability of the staff on a project. This is important because unexpected departures can affect the cost and schedule of a project. A too-high turnover rate means increased training and "learning curve" costs will be incurred as we replace the people who are leaving. This will usually have a cost and schedule impact.

**2.1.2.1 Graph**

The graph below shows unplanned turnover, as reported in August, for a specific project.



**2.1.2.2 Analysis and Discussion**

The unplanned turnover graph is plotted by month, using a line chart. It shows the rate at which people have left the project, by month, compared with the plan. Two lines are shown: the plan (black line) and the actual turnover (red line). When the red line is higher than the black line, it means more people are leaving than planned.

In the above example, there were minor fluctuations in the first part of the year but a significant increase in unplanned turnover in the summer months. The software manager has evaluated the cause of this high turnover rate and believes that the most probable cause is the failure of the air conditioning system and the resulting employee discontent. *<note that in this example, a possible reason is cited for the observed data. In some cases, all you can do is note what the graph is saying, give a list potential causes, and explain what actions are planned to determine the causes.>* The repair was not completed until late August. Corrective action recommended is to respond much more quickly to situations like this in the future. One approach would be to provide an emergency equipment repair budget for future years instead of relying on approval from the capital equipment vice president (who was on vacation at the time of the A/C failure). *<you can provide suggested solutions to the problem if you have identified a possible cause.>*

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

|  |
| --- |
| 2.1.2.3.1 Data Collection: Base Metrics Collected |
| The data required for computing unplanned turnover by project are:   * **Em** = the total number of employees on the project at the end of month **m** * **Lm** = the total number of employees who left the project during month **m** * **Rm** = the turnover rate expected (planned) for month **m**   The values of **Em** and **Lm** are collected monthly and recorded in separate rows in the data spreadsheet. All employees are counted and the numbers are recorded on the last day of each month.  The Planned turnover rate (**Rm**) is established by the project team leader. It is derived from historical turnover rates with revisions up or down to reflect actual experience (such as higher than average turnover that might be expected at certain times of the year or due to specific circumstances of this project). |

|  |
| --- |
| 2.1.2.3.2 Compound Metrics (Metrics Computed) |
| The Turnover rate **Tm** is defined for month **m** as follows:  **Tm = Lm / Em-1**  Unplanned turnover occurs any time **Tm > Rm**. |

|  |
| --- |
| 2.1.2.3.3 Data Refinement (manipulations, extractions, sorting, etc.) |
| The data must be checked for errors, and a spreadsheet row (“actual”) is created to compute the value of **Tm**, but little refinement or analysis is required beyond this. |

|  |
| --- |
| 2.1.2.3.4 How to Interpret the Graph |
| The horizontal axis of the graph shows the month and the vertical axis shows the turnover rate.  The Planned turnover **Rm** is stored in a "plan" row in the spreadsheet and can be revised as plans change. This generates the “Planned” line (black) in the graph.  The Actual turnover rate **Tm** is computed from **Em** and **Lm** and recorded in the “actual” row of the spreadsheet. This generates the “Actual” line (red) in the graph.  When the Actual line exceeds the Planned line by a significant amount or by more than one or two months, it indicates a problem that must be addressed. There is often a specific reason why so many employees are leaving, and it is important to determine that reason and take appropriate action as soon as possible. The other graphs (below) can often help in identifying the reasons for the high turnover rate. |

* + 1. **Unplanned Turnover for the Organization**

<continue as above>