Guideline for Risk Prediction Expert System

[**Overview**](#_30j0zll) **2**

[**I Environment Setup**](#_1fob9te) **2**

[Step 1: Install R (3.2.5)](#_3znysh7) 2

[Step 2: Install R packages](#_2et92p0) 2

[**II Data Preparation**](#_tyjcwt) **3**

[**III Train risk prediction models**](#_3dy6vkm) **4**

[Train Risk Prediction Model Based on Open Source Data](#_1t3h5sf) 4

[Step 1: Run the open-source risk prediction model training script.](#_4d34og8) 4

[Step 2: Check the output files.](#_2s8eyo1) 4

[Train Risk Prediction Model Based On USC-CSSE projects](#_17dp8vu) 5

[Step 1: Run the USC-CSSE risk prediction model training script.](#_3rdcrjn) 5

[Step 2: Check the output files.](#_26in1rg) 5

[**V Test the risk prediction models**](#_lnxbz9) **8**

[Step 1: Run the risk prediction model test script.](#_35nkun2) 8

[Step 2: Check the output files.](#_1ksv4uv) 8

[**VI Risk Prediction Model API Prototype**](#_z337ya) **10**

[Step 1: Land the risk prediction API page](#_3j2qqm3) 10

[Step 2: Input sample data](#_1y810tw) 10

[Step 3: Check output](#_4i7ojhp) 11

[**VII Data Collection**](#_2xcytpi) **11**

[Collecting Data for USC-CSSE Risk Prediction Model.](#_1ci93xb) 12

[Step1: Land on Risk Factors Setting Page using this url:](#_3whwml4) 12

[Step 2: Create a data sheet in .CSV format for your evaluation of the project](#_2bn6wsx) 13

[Collecting Data for Open Source Risk Prediction Model.](#_3as4poj) 13

[Step1: Find the required Jira reports and Github Commit History](#_1pxezwc) 13

[Step 2: Collect Phase and Issues Data](#_49x2ik5) 14

[Step 3: Collect time tracking data](#_2p2csry) 15

[Step 4: Collect code analysis data](#_147n2zr) 15

[Step 5: Collect accumulated bugs data](#_3o7alnk) 16

[Step 6: Collect Personnel Data](#_23ckvvd) 17

[Step 7: Assess Level of Risk](#_ihv636) 17

# Overview

This document aims at providing a tutorial and examples for the reader to access the operational details of risk prediction model.

Section I - VI give the tutorial and example based on the given data in the attached package.

Section VII provides instruction about where and how to collect data from your own data sources.

# I Environment Setup

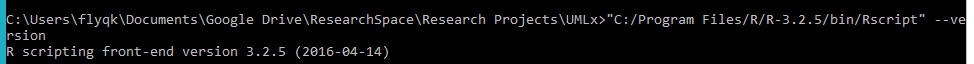
## Step 1: Install R (3.2.5)

1. Current experiment is based on R-3.2.5, which is available at:

<https://cran.r-project.org/bin/windows/base/old/3.2.5/>

1. Open Command Prompt of Windows OS, and type the following command to see if R is correctly installed.

*"C:/Program Files/R/R-3.2.5/bin/Rscript" --version*



## Step 2: Install R packages

1. Login R environment.
2. Type command:

*install.packages(“neuralnet”)*

1. Type command:

*install.packages(“ggplot2”)*

# II Data Preparation

In the attached package, the documents used as the inputs of the model training and testing processes are as follows:

1. The training dataset for the USC-CSSE risk prediction model:

"*Data/training\_data\_set\_usc\_model.csv*"

1. The training dataset for the Open Source risk prediction model:

"*Data/training\_data\_set\_open\_source\_model.cs*v"

1. The example input for testing the risk prediction model:

"*Data/Input\_Data\_Example\_5\_10.csv*"

# III Train risk prediction models

## Train Risk Prediction Model Based on Open Source Data

### Step 1: Run the open-source risk prediction model training script.

1. Open Command Prompt of Windows OS, and type:

*"C:/Program Files/R/R-3.2.5/bin/Rscript" ./Rscript/OpenSourceRiskPredicationModelTraining.R "Data/training\_data\_set\_open\_source\_model.csv"*

*\*Please follow the steps introduced in slides 33 - 40 of “Advanced Tollgate for 05-03-2018.pptx” or Section VII to create your own dataset.*

**

### Step 2: Check the output files.

1. Graphic representation file for the trained model:

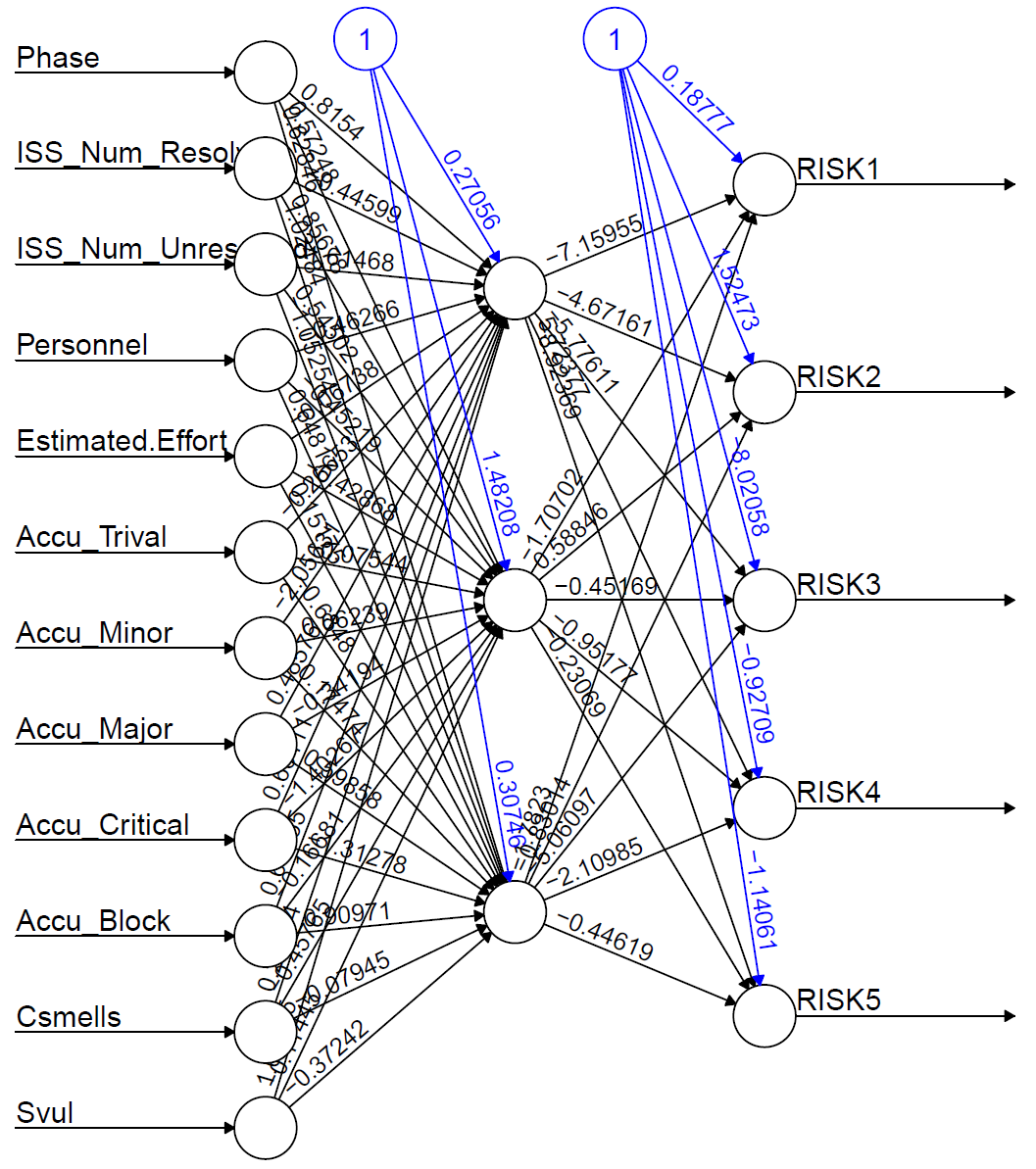
*“./Rplots2.pdf”.*

1. The model training report:

*“./Temp/open-source-risk-prediction-model-training-report.txt”*

1. The trained model:

*“./Model/riskPredictionModel\_open\_source.rds”*

**

*Figure 1. An example of the trained open source risk prediction model*

*\*If you run your data, the results will be produced in the same files but with different results.*

## Train Risk Prediction Model Based On USC-CSSE projects

### Step 1: Run the USC-CSSE risk prediction model training script.

1. Open Command Prompt of Windows, and type:

*"C:/Program Files/R/R-3.2.5/bin/Rscript" ./Rscript/ICSMRiskPredicationModelTraining.R "Data/training\_data\_set\_usc\_model.csv"*



\*Please follow the instructions in Section VII to collect the empirical data to train your own model.

### Step 2: Check the output files.

1. Graphic representation file for the trained model:

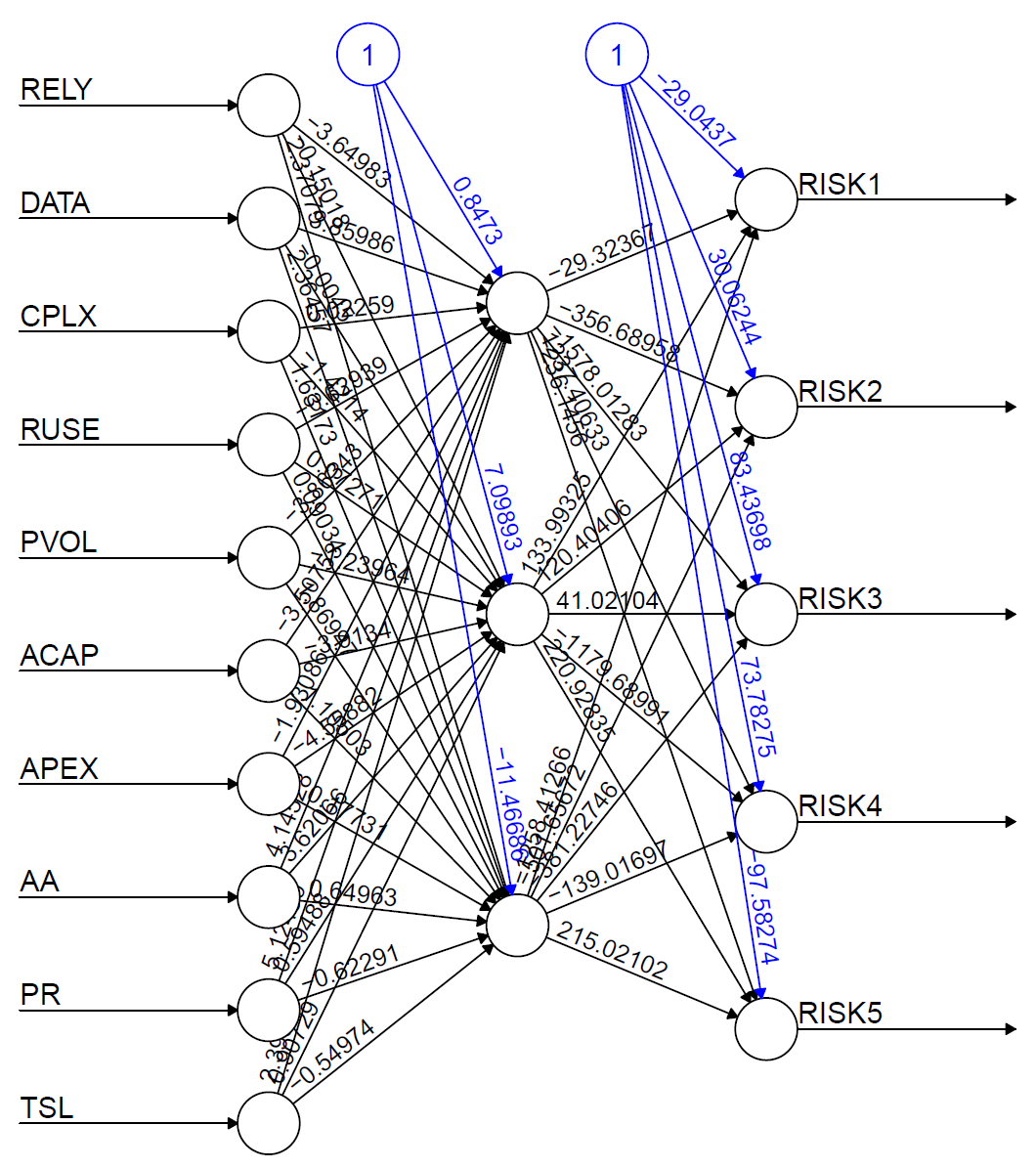
*“./Rplots3.pdf”*

1. The model training report:

*“./Temp/icsm-risk-prediction-model-training-report.txt”*

1. The trained model:

*“./Model/riskPredictionMode\_icsm\_projects.rds”*

**

*Figure 2. an example of trained USC-CSSE risk prediction model*

IV Run the trained model for risk prediction

Step 1: Run the risk prediction model test script.

1. Open Command Prompt of Windows, and type:

*"C:/Program Files/R/R-3.2.5/bin/Rscript" ./Rscript/RiskPredication.R "Data/Input\_Data\_Example\_5\_10.csv"*

*\*Please follow the steps introduced in slides 33 - 40 of “Advanced Tollgate for 05-03-2018.pptx” or Section VII to create your own dataset.*



Step 2: Check the output files.

1. The risk prediction report:

*“./Data/risk-prediction-report.txt”*

|  |
| --- |
| *An example of the “risk-prediction-report.txt” is as follows:*     1. Is the predicted probabilities of the five levels of risk based on the ICSM risk prediction model. 2. Is the predicted probabilities of the five levels of risk based on the open source risk prediction. 3. Bayesian averaged estimates of the probabilities for the five levels of risk. We choose the most probable as the final estimate. |

2. The risk prediction report:

*“./Temp/risk-prediction-results.txt”*

|  |
| --- |
| The simplified output of the risk prediction results, which is used for the risk prediction api. |

# V Test the risk prediction models

## Step 1: Run the risk prediction model test script.

1. Open Command Prompt of Windows, and type:

*"C:/Program Files/R/R-3.2.5/bin/Rscript" ./Rscript/RiskPredicationModelTestingBootstrap.R "Data/training\_data\_set\_usc\_model.csv" "Data/training\_data\_set\_open\_source\_model.csv"*

## Step 2: Check the output files.

1. The testing report by bootstrapping:

*“./Temp/risk-prediction-model-testing-report.txt”*

|  |
| --- |
| ***An example of the “risk-prediction-model-texting-report.txt” is as follows:***   1. The risk prediction accuracy for the ICSM risk prediction model (1000 resampling for bootstrapping estimate).     2. The risk prediction accuracy for the open source risk prediction model (1000 resampling for bootstrapping estimate).    3. The risk prediction accuracy for the combined risk prediction model (1000 resampling for bootstrapping estimate). |

# 

# 

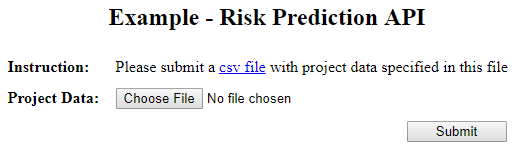
# VI Risk Prediction Model API Prototype

This section gives a quick example about how to use a sample data point to request for its risk level from a Amazon EC2 Service, on which we deployed the current risk prediction model.

## Step 1: Land the risk prediction API page

Go to [this link](http://ec2-54-67-99-52.us-west-1.compute.amazonaws.com:8081) or type the following address on the browser to load the Risk Prediction API Prototype Page:

[*http://ec2-54-67-99-52.us-west-1.compute.amazonaws.com:808*](http://ec2-54-67-99-52.us-west-1.compute.amazonaws.com:8081)*1*



## Step 2: Input sample data

As the input, upload a csv file including the defined factors (Refer to Slide#10 - 21 ).



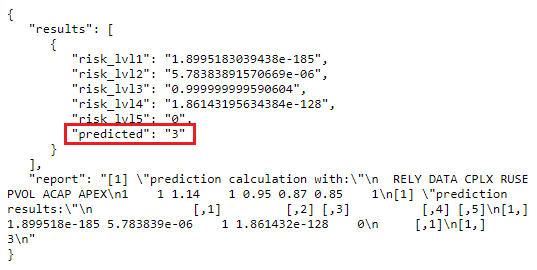
A sample input file can be downloaded from the prototype page. It includes the following factors:

Table 1. Factors for Risk Prediction Models

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USC-CSSE Risk Prediction Model** | |  | **Open Source Risk Prediction Model** | |
| **Factor** | **Description** |  | **Factor** | **Description** |
| RELY | Required Software Reliability |  | Phase | # of Phase |
| DATA | Database Size |  | ISS\_Num\_Resolved | # of resolved issues |
| CPLX | Product Complexity |  | ISS\_Num\_Unresolved | # of unresolved issues |
| RUSE | Developed for Reusability |  | Personnel | # of contributors |
| PVOL | Platform Volatility |  | Estimated\_Effort | Total estimated effort |
| ACAP | Analyst Capability |  | Accu\_Trivial | # of accumulated trivial issues |
| APEX | Applications Experience |  | Accu\_Minor | # of accumulated minor issues |
| AA | Level of Automated Analysis |  | Accu\_Major | # of accumulated major issues |
| PR | Level of Peer Review |  | Accu\_Critical | # of accumulated critical issues |
| TSL | Level of Test Sophistication |  | Accu\_Block | # of accumulated block issues |
|  |  |  | CSmell | Code Smells |
|  |  |  | SVul | Code Security Vulnerability |

## Step 3: Check output

Once uploaded, the output (i.e. the result of risk prediction), in JSON format, given by the model, will be shown in seconds.



For example, on the above screenshot, the output lists the possibility of each Risk Level (Refer to Slide# 25 - 26) based on the input factors, and Risk Level 3 is with highest possibility. Therefore the predicted risk level is 3. The output also prints out a brief report about the input factors, as needed, for further insights.

# VII Data Collection

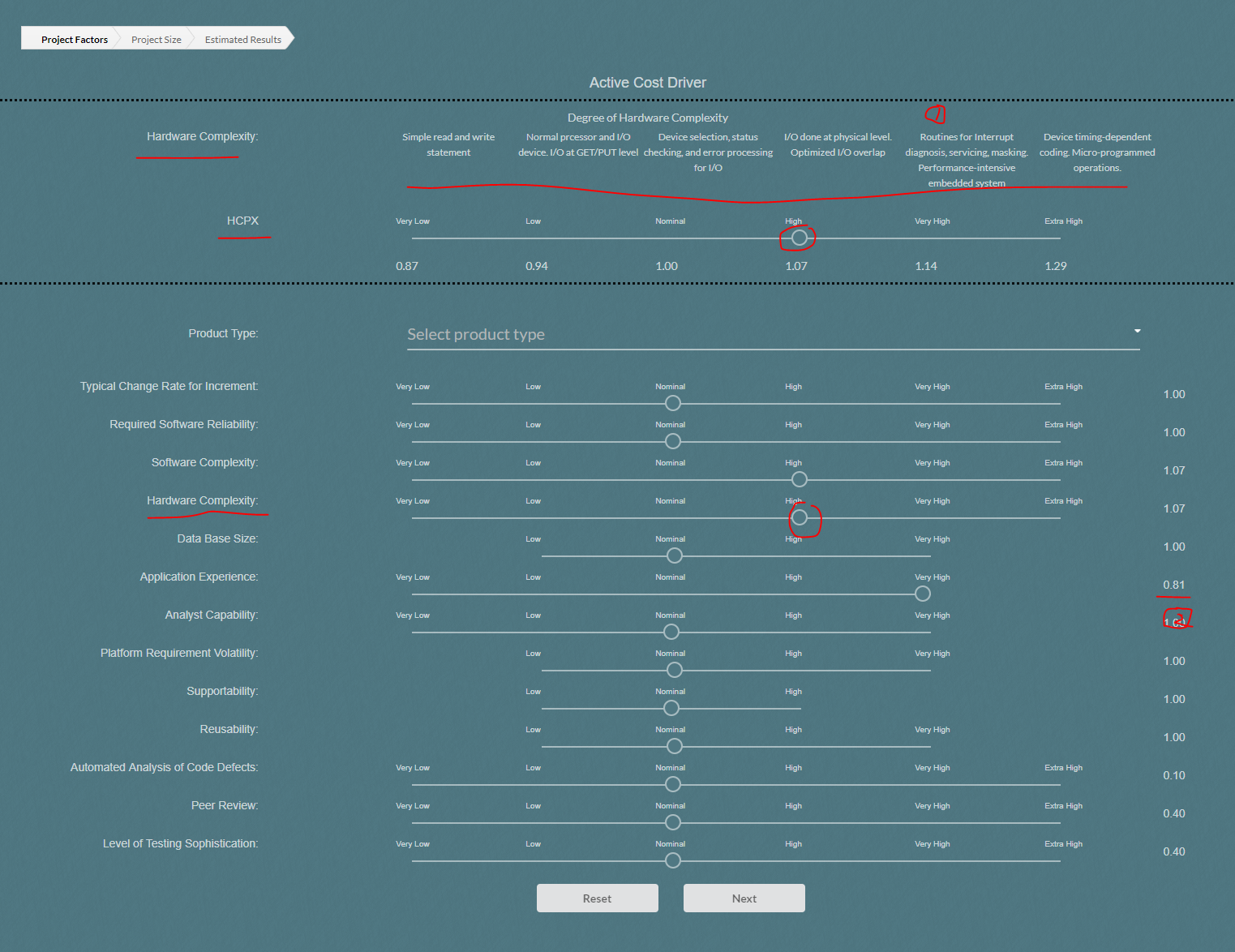
This section provides instructions about where and how to collect data from your own data sources. Specifically data corresponding to the factors listed in Table 1 need to be collected to train your own risk prediction models.

## **Collecting Data for USC-CSSE Risk Prediction Model.**

### Step1: Land on Risk Factors Setting Page using this url:

<http://ec2-54-67-99-52.us-west-1.compute.amazonaws.com:8686/demo/phase5/start/factorsInputPage.html>

This page provides the definitions (1) for factors. Evaluate your project from extra low - extra high for the factors and find their ratings (2).



*Figure 3. Expert system prototype for risk evaluation.*

### Step 2: Create a data sheet in .CSV format for your evaluation of the project

### 

*Figure 4. Example of collected data for USC-USSE risk prediction model.*

## **Collecting Data for Open Source Risk Prediction Model.**

### Step1: Find the required Jira reports and Github Commit History

1. Jira Repos

I. Release and Milestones.

Related factors in prediction model: ***Phase***

II. Effort tracking report.

Related factors in prediction model: ***Effort*** (actual effort), ***Estimated\_Effort*** (estimated effort)

III. Issue tracking reports: issue creation and resolution reports.

Related factors in prediction model: ***ISS\_Num\_Resolved, ISS\_Num\_Unresolved, Accu\_Trivial, Accu\_Minor, Accu\_Major, Accu\_Critical, Accu\_Block***

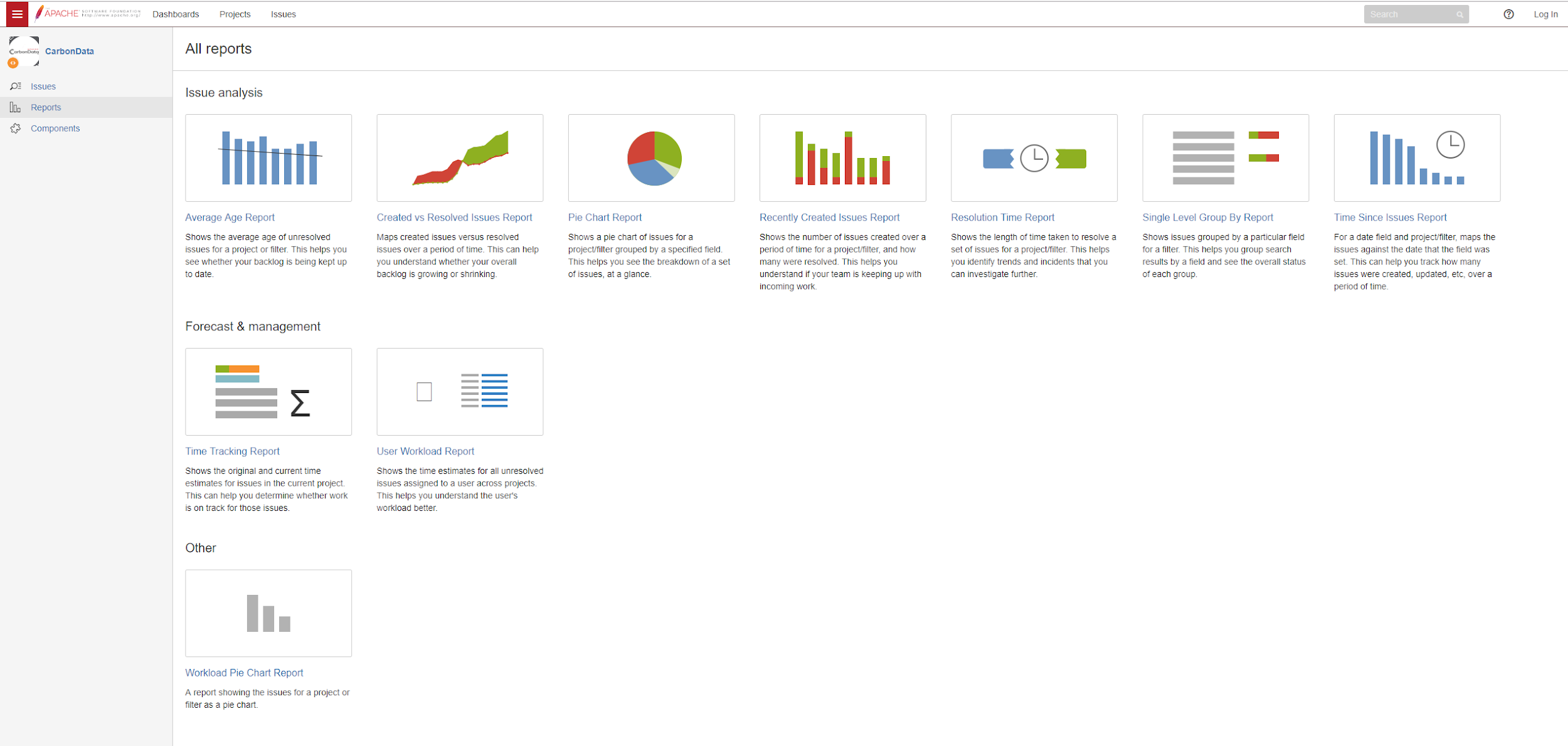
IV. Personnel and Contributions.

Related factors in prediction model: ***Personnel***

V. System Modules.

\*An example of Jira reports can be found at Url:

*https://issues.apache.org/jira/projects/CARBONDATA?selectedItem=com.atlassian.jira.jira-projects-plugin:report-page*



*Figure 5. Jira reports panel to download related datasheets.*

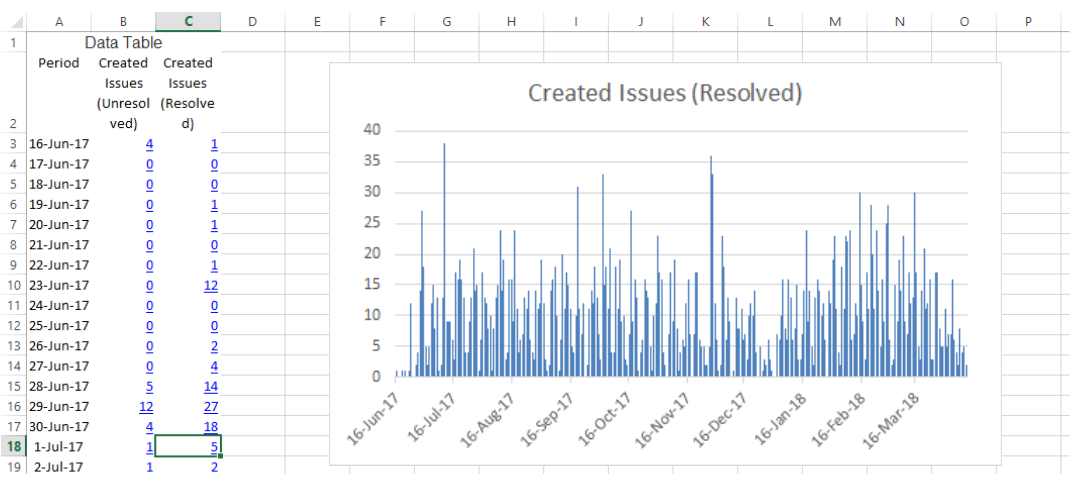
2.Github

I.Commits.

Related factors in prediction model: ***CSmell, SVul*** *(CSmell and SVul can be drived using USC-CSSE’s SQUAAD tool or SonarQube)****.***

### Step 2: Collect Phase and Issues Data

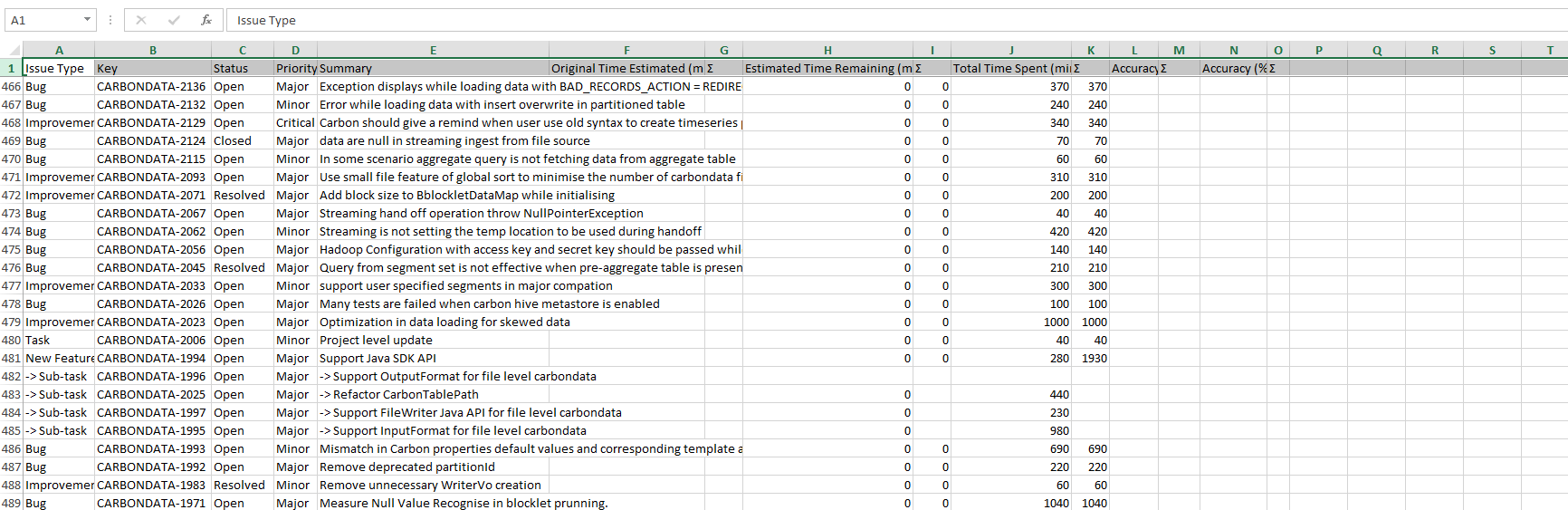
1. The distribution of creation time indicates the phases - ***Phase***. (If actual definition of phases is available, it would be better)
2. Issue records provides creation times and resolution times of the issues. ***ISS\_Num\_Resolved, ISS\_Num\_Unresolved*** are calculated by ***Phase*** using the issue report

******

*Figure 6. Example of issue related datasheet.*

### Step 3: Collect time tracking data

Time tracking records provide estimated times and actual times spent on certain tasks.***Effort*** (actual effort), ***Estimated\_Effort*** (estimated effort) for each ***Phase*** is calculated by the effort spent on the issues that belong to the ***Phase***.



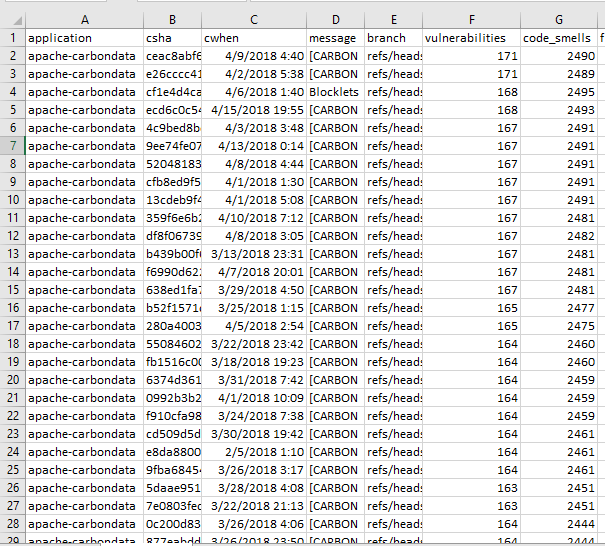
*Figure 7. Example of time tracking datasheet.*

### Step 4: Collect code analysis data

Code metrics applied to each commit to measure code quality and technical debt.

* Number of Code smells (***CSmell***) is calculated by commits that belong to ***Phase***
* Number of Vulnerabilities (***SVul***) is calculated
* by commits that belong to ***Phase***

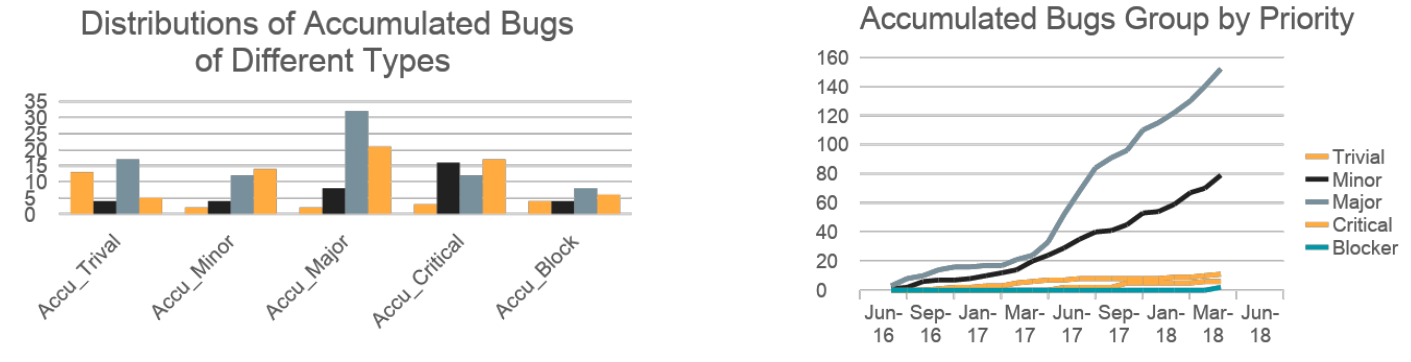
\* This analytical data is generated by SQUAAD (or SonarQube)



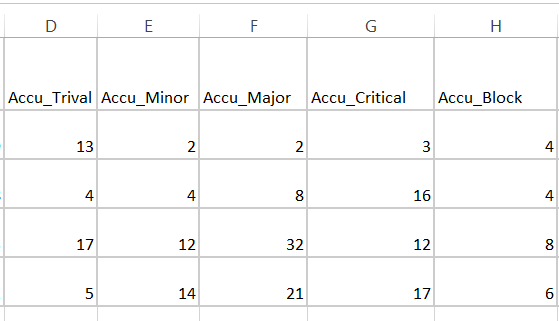
*Figure 7. Example of commits datasheet.*

### Step 5: Collect accumulated bugs data

Accumulated Bugs of different severity reflect the potential technical debt. ***Accu\_Trivial, Accu\_Minor, Accu\_Major, Accu\_Critical, Accu\_Block*** are determined by categorizing the bugs according to their severity.



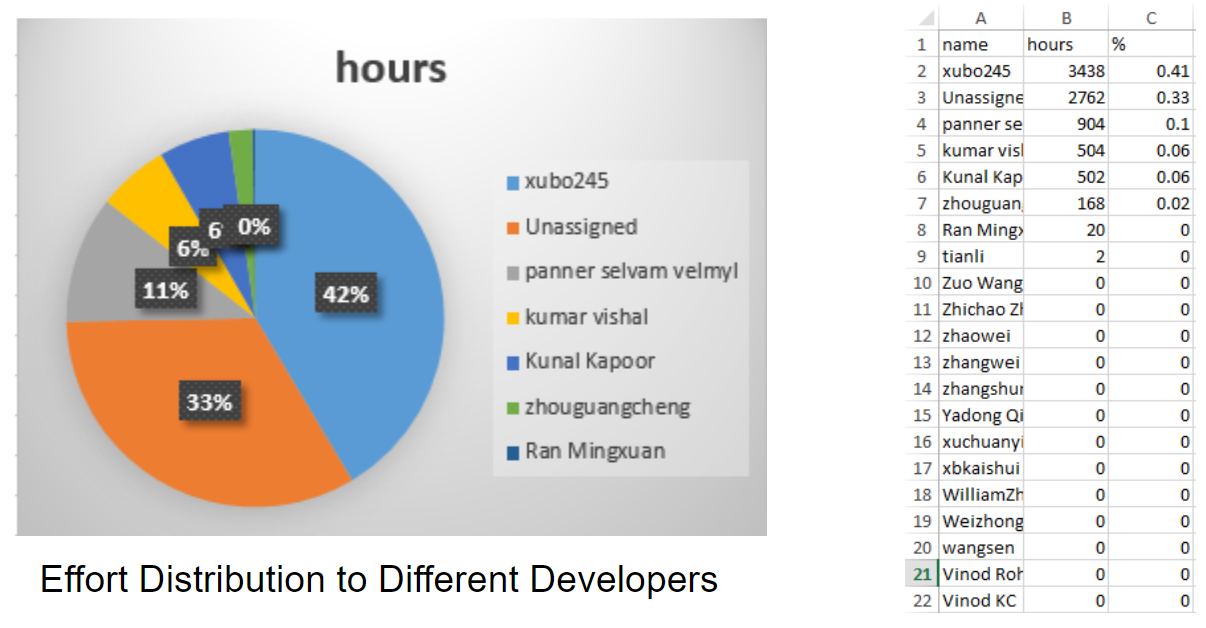
*Figure 7. Example of distributions of categorized bugs.*



*Figure 8. Example of categorized bugs based on severity.*

### Step 6: Collect Personnel Data

Effort distribution over team members identify the contributors, which helps measure the balance of workload. The distribution of the effort to personnel determines the core contributors, the number of which define the factor ***Personnel****.* Contributors who contribute larger than 5% are determined as core contributors.

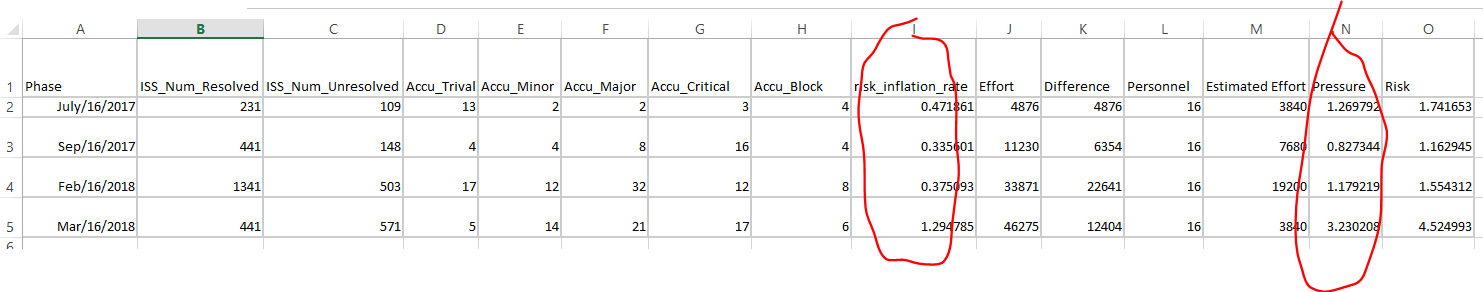


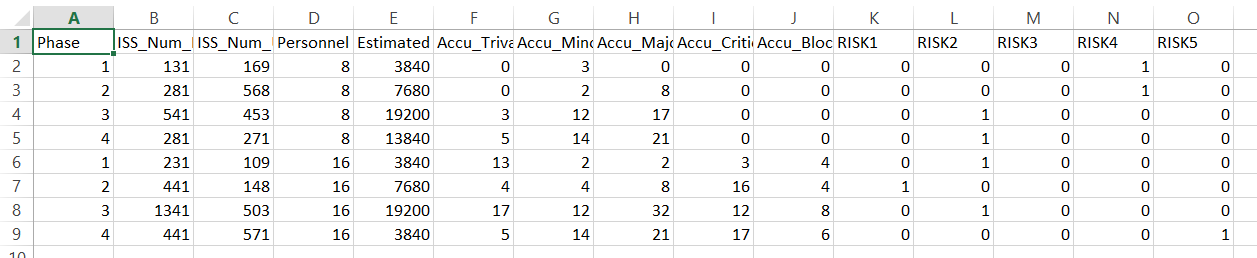
*Figure 9. Example of personnel distribution chart and datasheet.*

### Step 7: Assess Level of Risk

The proposed method to measure risk.

1. ***Risk\_inflation\_rate*** = ***ISS\_Num\_Unresolved*** (number of unsolved issues) / ***ISS\_Num\_Resolved*** (number of solved issues) (at the end of a milestone)
2. ***Pressure*** = ***Actual\_Effort*** / ***Estimated\_Effort*** (for each phase)
3. ***Risk*** = ***Risk\_inflation\_rate*** + ***Pressure***

*Figure 10. Example of datasheet for risk evaluation for open source risk prediction model.*

Step 8: Prepare the data sheet (.csv) using the collected data 

*Figure 11. Example of input for open source risk prediction model.*

\*After collect the data for USC risk prediction model and Open source risk prediction model, you can use the procedures introduced in Section III to train your own risk prediction model.