

Hourglass Bug Predictor



User Guide

V1.0

Contact: support@hourglass-software.com

About

Hourglass Bug Predictor is a Command Line Tool that integrates with Jira and uses Machine Learning to predict the future quality of your products in development. It will predict how many bugs you should expect in the next seven days, from the time the software is run.

It will query your Jira tracking system for Bugs under a specified Jira “Project”. Based on the data it finds in your Jira Management System, it will train a Machine Learning Model and then predict the number of bugs you should expect to find in the following week of your development.

For example, it will tell you/predict that there will be 5 bugs in the next week or some other prediction value. The prediction takes into account the fields of your Jira Bugs both for training as well as prediction. The prediction feature vector also takes historical timelines and grouping of your bug data and respective attributes (see details below).

This will help you gauge the quality of your systems.

Example Output

Bug Prediction Count for next 7 days for input parameters:0.5

Pre-requisites

1. Jira host/endpoint
2. Java 8
3. Windows, MacOS, or Linux
4. Jira API token and username
5. Jira Project & Bugs history
6. (Product under development as target)

Setup

1. **Download the ZIP file** from <https://hourglass-software.s3.amazonaws.com/Hourglass+Bug+Predictor.zip>
2. Windows
 - a. Download the ZIP file and extract it in the root of you C:\ drive (ie: C:\Hourglass Bug Predictor)
3. MacOS
 - a. Extract the zip file to the root of /Users/ directory (ie: /Users/Hourglass Bug Predictor)
4. DO NOT DELETE ANY FILES OR FOLDERS
 - a. The Application binary classes will be in the \bin folder
 - b. Runtime dependencies will be in the \dependencies folder (do not delete any)
5. The ZIP file includes a sample input schema file and a sample input parameter data file (see details below)
6. Make sure you have Java (8) bin in your PATH environment variables and JAVA_HOME
7. (NOTE: to change location of #1 or #2, you must update the dependencies path in the .bat or .sh files)

Usage

1. Once you've extracted the zip file to the root of your C: drive, open **Windows Powershell or shell/terminal** to go to the \Hourglass Bug Predictor directory.
2. Run the run-prediction.bat file or run-prediction.sh
3. For the .bat file, use the following parameters command line arguments:
 - a. Jira url
 - b. Jira username/email address for login
 - c. Jira API Token (from your Jira software access settings)
 - d. Your bug items *schema .xml file* (file should be in the root of C:\Hourglass Bug Predictor\
 - i. (See details below)
 - e. Your *input parameters for prediction control .csv file* (file should be in the root of C:\Hourglass Bug Predictor\
 - i. (see details below)

For example:

```
.\run-prediction.bat https://hourglass-software.atlassian.net faisal@hourglass-software.com  
7khWRw976PKYJvOuAnP16F53 sample-bug-predictor-schema.xml sample-predictParamData.csv
```

Schema XML File

The schema xml file is to tell the software how your Jira Bugs are structures and what fields they use (for querying Jira, parsing the responses, and creating training data).

The schema XML file root element is <bugpredictor>. And then has the following fields with additional definitions of the fields:

<project> - *this tells the software which Jira Project to query*

Example:

```
<project>HBP</project>
```

<components> - *provide the components your Bugs use*

Example:

```
<components>
    <component>test1</component>
    <component>test2</component>
    <component>test3</component>
</components>
```

<priorities> - *priorities of your bugs. Can be Jira default or custom, but should be defined here*

Example:

```
<priorities>
    <priority>Lowest</priority>
    <priority>Low</priority>
    <priority>Medium</priority>
    <priority>High</priority>
    <priority>Highest</priority>
</priorities>
```

<labels> - *the labels you use for your bugs*

Example:

```
<labels>
    <label>label1</label>
    <label>label2</label>
```

```
<label>label3</label>
```

```
</labels>
```

<customFieldName> - *this is OPTIONAL (with children elements). If your Bugs use a custom field that you would like to include for prediction. Define it here. Only 1 custom field definition is allow (but with as many possible values)*

Example:

```
<customfields>
```

```
    <customfield>cf1</customfield>
```

```
    <customfield>cf2</customfield>
```

```
    <customfield>cf3</customfield>
```

```
</customfields>
```

Full XML file example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<bugpredictor>
```

```
    <project>HBP</project>
```

```
    <components>
```

```
        <component>test1</component>
```

```
        <component>test2</component>
```

```
        <component>test3</component>
```

```
    </components>
```

```
    <priorities>
```

```
        <priority>Lowest</priority>
```

```
        <priority>Low</priority>
```

```
        <priority>Medium</priority>
```

```
        <priority>High</priority>
```

```
        <priority>Highest</priority>
```

```
    </priorities>
```

```
</labels>
```

```
        <label>label1</label>
        <label>label2</label>
        <label>label3</label>
    </labels>
    <customFieldName>customfield</customFieldName>
    <customfields>
        <customfield>cf1</customfield>
        <customfield>cf2</customfield>
        <customfield>cf3</customfield>
    </customfields>
</bugpredictor>
```

You can modify this file for different projects before any execution of Hourglass Bug Predictor.

Input Parameter CSV File for Prediction Control

This file allows you to make predictions of the number of bugs to expect with certain field values. For example, you can predict High Priority Bugs with Component A and Label B. The prediction will tell you the number of expected Bugs to find with those attributes. You can modify these parameters for different executions to find different predictions based on field values. The different types of parameters are on separate lines of the file. For example, component to use in the first, labels to use in the second, priority to use in the third, and custom field value to use (if applicable) on the last.

The format of the .csv file is (no spaces allow between labels values, only comma):

Component name to predict for

Comma separated list of label values to predict for (or just one value)

Priority of Bug to predict for

Custom field value to predict for (if used as is optional)

For example:

test1

label1,label2

Medium

cf1

LICENSE

On purchase for \$9.99, a single user can use this software unlimitedly. You may not redistribute or resell this software. Additionally, you may not reverse assemble or decompile the binary. Any breach found of these restrictions will pursued to the full extent of the law.