# EECS 4313 Assignment 1
## - Bugs and Bug Reports

## 1. Assignment goals

The purpose of this assignment is to give you experience reporting bugs and developing test cases. This assignment will give you practice thinking how to develop a test case from a requirements document, what a professional bug report should be, as well as analyzing bugs before reporting them.

## 2. [Task 1] Reporting Bugs or Proposing Enhancements

For this task, you will work with the BORG open source applications.

### 2.1 Getting started

- Download the BORG Calendar installation package (https://github.com/mikeberger/borg_calendar/releases). Pick the latest release (**version 1.9.0.4**) to install. For this assignment, you can choose either the jar file or the exe file to install. Run it and follow the on-screen instructions. *Remember your installation path!*
  - o Avoid Linux as your testing environment, as it is reported that you might run into installation and configuration issues.
- Explore the product website and check out the BORG Calendar product specification.
  - o http://www.cse.yorku.ca/~zmjiang/teaching/eecs4313/assignments/SRSforBORGCalendar.pdf

### 2.2 Bug reports

You should create a written report, which contains 3 bug reports. These bug reports can either be reporting bugs or proposing enhancements. Among these 3 bug reports, you should report at least 2 bugs and at most 1 feature enhancement. *Please do not submit these bug reports to the project website's issue reporting system!* You can refer to the Appendix A for the formatting of the bug reports.

In the bug reports, you need to describe the steps to reproduce the bugs and any follow-up testing you did to strengthen your bug report. Since the marker may be unfamiliar with the functionality tested, you must explain what the expected functionality was and how you decided that your test case resulted in a failure. Include relevant screen shots if applicable.

Since you will probably not have time or the resources to run many follow-up tests, describe other tests which you would like to run. You should provide justification of these follow-up tests and explain what you would hope to learn from them.

For each report focus on the following:

- Clarity and tone.
- Attempt to replicate the bug.

- Perform further analysis using various types of follow-up testing.
- Include important information that you successfully replicated the bug on XXX configuration in YYY build, or describing a simpler set of replication steps (if you found one). Your comments should NEVER appear critical or disrespectful of the developer.

Issues to consider when working on a bug report:

- Is the summary short (about 50-70 characters) and descriptive?
- Can a developer understand the bug report? Is there sufficient detail to envision what the program did in response? Is it clear what the failure was?
- Is it obvious where to start (what state to bring the program to) to replicate the bug?
- Is it obvious what files to use (if any)? Is it obvious what you would type?
- Is the replication sequence provided as a numbered set of steps, which tell developer exactly what to do and, when useful, what developer will see?
- Does your bug report include unnecessary information, personal opinions or anecdotes that seem out of place?
- Is the bug report too long? Too short? Does it have a lot of unnecessary steps?
- Can you replicate the bug by following your steps?
- Can developer get lost or wonder whether you had done a step correctly? Would additional feedback, e.g. "the program will respond like this ...", have helped?
- Does configuration or environment change have an effect on bug reproduction?

# 3. [Task 2] Existing Bug Reports in Open Source Systems

Open source software engineers use a set of development tools while they develop, test and maintain their software systems. For example, they use source code repositories (e.g., SVN or Git) to archive their historical revisions and use bug tracking systems (e.g., Bugzilla or Jira) to report, communicate and fix their issues. In this task, you are asked to evaluate the quality of existing bug reports in some of the popular open source systems.

## 3.1 Bug Reports in Mozilla Firefox

The Mozilla Firefox project uses the BugZilla issue reporting system. In this sub-task, please read Mozilla bug report 112785 (https://bugzilla.mozilla.org/show_bug.cgi?id=112785) and point out any problems with the initial bug report (as seen in the "Title" and the "Description").

## 3.2 Issue Reports in HBase

In this task, you are provided with the bug reports of HBase (https://hbase.apache.org/). HBase is a popular open source NoSQL database, modeled after Google's BigTable (https://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf). Different from Firefox, HBase uses Jira as its issue tracking system.

You are given a zip file called "hbaseBugReport.zip". There are many XML files inside, each of which corresponds to one bug report. For example, the bug report "bugReport_11076.xml" corresponds to

HBase bug report (https://issues.apache.org/jira/browse/HBASE-11076). Suppose you were the software engineering manager of HBase and you are interested in finding out the status about the project: [Make sure you explain clearly in the report how you obtained the above results.]

1. There are various kinds of issue reports in HBase (e.g., bugs, enhancements, tasks, etc.). Each issue report can be in different stages (e.g., open, closed, fixed, etc.) Among all these issue reports, which belong to the "bug" type, how many of them are there in total?

2. Among the total number of "bugs", give a quantitative report on their current state. In other words, report the number of bug reports which are in different states (a.k.a., open, closed, or fixed, etc.).

3. Among the "bug type" issue reports, which are either "fixed" or "closed", what are minimum, average, median, and maximum bug resolution time?

## 4. Submission

You must work in a team with **at least two and at most six** partners. *Failure to form a team or a team with the right sizes will result in loss of marks!* Submit a PDF of your report (a1.pdf) electronically. Your report must include the names and the student numbers of all team members.

Use the following command to submit:

- submit 4313 a1 a1.pdf

# Appendix A – Report Format

Your report should be single space 12 point font. Each of the bug report should include the following information:

**Bug Report Title:**
About 3 to 10 words making clear what your bug report is about.

**Reported by:**
The bug reporter's name and contact information.

**Date reported:**
The date when this bug report is filed.

**Program (or component) name:**
Which program/component has issue?

**Configuration(s):**
The hardware and software configurations under which the bug was found and replicated.

**Report type:**
Is this a bug (e.g., coding error, design issue or documentation mismatch) or feature enhancement?

**Reproducibility:**
Yes / no / sometimes / unknown. For no/sometimes, provide as much information as you can.

**Severity:**
Is this a major issue or a minor issue?

**Problem summary:**
A short summary of the problem

**Problem description:**
Describe the problem and the steps to reproduce the problem.

**New or old bug:**
Is this a new bug or an existing unfixed bug?