SE465
Software Testing, Quality Assurance, and Maintenance
Assignment 1 Technical Notes

Patrick Lam
Release Date: January 17, 2019

This document explains how to set up a working environment for Assignment 1. I've chosen to use Vagrant plus glitch to make it easy to set up your environment. I've tested these instructions on Debian GNU/Linux as well as Windows. They should work on a Mac as well.

Initializing your virtual machine

Install the following software:

- you should already have git, since you cloned the `a1` repository;
- virtualbox (`https://www.virtualbox.org/wiki/Downloads`); you don't need the extension pack or SDK;
- vagrant (`https://www.vagrantup.com/downloads.html`).

You should have an `se465-1191-USERNAME-a1` directory after cloning your provided git repository. In that directory, you will find a `Vagrantfile`, `bootstrap.sh`, along with what you need for Q1 and templates for your answers.

Next, you need to get vagrant to build your virtual machine.

- Go to the `a1` subdirectory, and

  ```
  $ vagrant up
  ```

  This initializes your virtual machine and downloads the `average` sample code into the virtual machine.
  potential pitfall: you may get a cryptic error about "VT-x not available". In that case, you need to go to your computer's BIOS settings and enable virtualization extensions. (See `http://superuser.com/questions/22915/how-do-i-enable-vt-x` for information.)

- Start an `ssh` seesion into the virtual machine you've just set up:

  ```
  $ vagrant ssh
  ```

  potential pitfall: `ssh` may not be set up/in the PATH on your (Windows) computer. Either put it there (git includes ssh), or ssh directly into your virtual machine:

  ```
  > ssh vagrant@localhost -p 2222 -i <address-vagrant-ssh-tells-you>
  ```

Great! Now you have a working virtual machine.

Editing files

The Vagrant configuration is set up such that the `shared` directory is also visible inside the virtual machine in your home directory there. You can use your favourite text editor on your host machine, or you can install vim or Emacs inside the VM and edit there. Because the directory is shared, committing and pushing your clone of the repository from your host machine will send us your submission.