CS 521/621
Homework 2
Pair Programming

Due: **Oct 7, 2013, 9:00 AM EDT** via Moodle. You are encouraged to work in pairs on this assignment, but you may work alone if you wish. You may not work in groups larger than 2. And each student must submit his or her own write up, clearly specifying the collaborators. The write ups should be individual, not created jointly, and written in the student's own words. Late assignments will not be accepted without **prior** permission.

Overview

This is a creating assignment. The goal of this assignment is to observe pair programming, and then brainstorm what aspects of the pair programming process could be automated.

The assignment consists of:

1. Watching **three** episodes of a peer programming show and keeping a careful log of all times the developers do something wrong, discover something they did wrong, or fix something they did wrong.
2. Writing a 1-page summary of your ideas of what can be automated about the pair programming process.

Resources

- A random number generator, such as `http://www.random.org/integers/`
- Pair programming videos: `http://pairwith.us/tv`

Part I

As software engineer research aims to help developers, it often makes sense to observe developers and note the problems they are having. One kind of research is identifying the problems developers need help with, and how various tools would help them. To help us do this, we often record developers in action, and the analyze the videos.

You will need to watch **three consecutive** episodes of the pair programming videos. Let's stick to picking from just the first 10 episodes for now, so use a random number generate to pick a random integer x between 1 and 8 (inclusive), and then watch episodes x, $x+1$, and $x+2$.

While watching the episodes, your job is to record three kinds of events:

1. Developers create a defect (but don't necessarily know this yet).
2. Developers discover a defect.
3. Developers fix a defect.

A defect can be a lot of things. Basically, a defect is any kind of mistake or anything the developers did wrong. It could be compilation error, something that breaks a test, a design decision error, a communication error, an error in a diagram, etc.

The videos you will be watching are especially great for catching defects because they show developers pair programming, so they are talking out loud about what they are doing (and making jokes). The talking really helps understand what they are doing, and what they are thinking, and when they make mistakes. You can identify the defects when a tool (e.g., a compiler or some other static analysis tool) says there is a defect, or when the developers themselves say they found something wrong. You'll then often have to backtrack in the video to find when the developers introduced the defect. Sometimes, the defects will be discovered before the developers have written any code; that's important too.

In addition to the `when created`, `when discovered`, and `when fixed` times, for each defect, you will classify the defects by what kind of tool can find them.

You should feel free to be very creative in this process. Look for things that went wrong that could have been fixed with some amazing tool that doesn't exist yet. Don't be shy about identifying defects even if you are unsure if they are really full-out defects; maybe it's not something going wrong but just a slight derailment for the developers, but it's still something we could potentially help them with.

Use a spreadsheet to store your observations. Get a copy of a `.xls` spreadsheet with the proper headings here: `http://cs.umass.edu/~brun/class/2013Fall/CS521.621/hw2/pairProgrammingObservations.xls`. You may use a spreadsheet editor of your choice, such as Excel, numbers or (free) Google Docs. Most of them will be able to open and save `.xls` files. Submit your answers in this file, and name it `pairProgrammingObservations`*`YourName`*`.xls`, (replacing *`YourName`* with, well, your full name, in the LastFirst format.

Part II

After watching the videos, you will be inspired by all the ideas of the tools that you could make to help people discover or avoid mistakes while programming. Organize your thoughts and create a 1-page write-up describing one idea (or a couple if you're very good at writing clearly and concisely enough to describe two ideas in 1 page) for such a tool.

Be clear about:

- the kinds of issues this tool can fix,
- how the tool will be able to do this,
- how would you evaluate this tool to show everyone how great it works.

If you are feeling stuck, consider what parts of the navigator can be automated, and how.

Submit your 1-page write up as a `.pdf` named `pairProgramming`*`YourName`*`.pdf`, (replacing *`YourName`* with your full name, in the LastFirst format.

Deliverables

Submit two files:

1. The `pairProgrammingObservations`*`YourName`*`.xls` file of your video observations.
2. Your `pairProgramming`*`YourName`*`.pdf` write-up.