2769

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
SUBMISSION FOR NEW STUDIES

GENERAL INFORMATION

1.	**Date Submitted:** _____
1a.	**Revised Date:** _____
2.	**Title of Project:** *Interactive Bug Fixing*
3.	**Principal Investigators:** *Yoonki Song (Ph.D. Student)*
4.	**Department:** *Computer Science*
5.	**Campus Box Number:** *8206*
6.	**Email:** *ysong2@ncsu.edu*
7.	**Phone Number:** *919-673-9194*
8.	**Fax Number:** _____
9.	**Faculty Sponsor Name and Email Address if Student Submission:** *Dr. Emerson Murphy-Hill; emerson@csc.ncsu.edu*
10.	**Source of Funding? (required information):** *Google*
11.	**Is this research receiving federal funding?:** *No*
12.	**If Externally funded, include sponsor name and university account number:** _____
13.	**RANK:** ☐ Faculty ☒ Student: ☐Undergraduate; ☐ Masters; or ☒PhD ☐ Other (specify): _____

As the principal investigator, my signature testifies that I have read and understood the University Policy and Procedures for the Use of Human Subjects in Research. I assure the Committee that all procedures performed under this project will be conducted exactly as outlined in the Proposal Narrative and that any modification to this protocol will be submitted to the Committee in the form of an amendment for its approval prior to implementation.

Principal Investigators:

Yoonki Song (typed/printed name) — (signature) * — *7/25/2012* (date)

*As the faculty sponsor, my signature testifies that I have reviewed this application thoroughly and will oversee the research in its entirety. I hereby acknowledge my role as the **principal investigator of record**.*

Faculty Sponsor:

Emerson Murphy-Hill (typed/printed name) — (signature) * — *7/25/2012* (date)

***Electronic submissions to the IRB are considered signed via an electronic signature. For student submissions this means that the faculty sponsor has reviewed the proposal prior to it being submitted and is copied on the submission.**

Please complete this application and email as an attachment to: debra_paxton@ncsu.edu or send by mail to: Institutional Review Board, Box 7514, NCSU Campus (Administrative Services III). **Please include consent forms and other study documents with your application and submit as one document.**

****

For SPARCS office use only

Reviewer Decision (Expedited or Exempt Review)

☑ Exempt b2 ☐ Approved ☐ Approved pending modifications ☐ Table

Expedited Review Category: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8a ☐ 8b ☐ 8c ☐ 9

Reviewer Name — Signature — 8-24-12 Date

North Carolina State University is a land-grant university and a constituent institution of the University of North Carolina

Office of Research and Innovation
Division of Research Administration

NC STATE UNIVERSITY

Campus Box 7514
Raleigh, North Carolina 27695-7514

919.515.2444 (phone)
919.515.7721 (fax)

From: Deb Paxton, IRB Administrator
North Carolina State University
Institutional Review Board

Date: August 27, 2012

Title: Interactive Bug Fixing

IRB#: 2769

Dear Yoonki Song

The research proposal named above has received administrative review and has been approved as exempt from the policy as outlined in the Code of Federal Regulations (Exemption: 46.101. b.2). Provided that the only participation of the subjects is as described in the proposal narrative, this project is exempt from further review.

NOTE:

1. This committee complies with requirements found in Title 45 part 46 of The Code of Federal Regulations. For NCSU projects, the Assurance Number is: FWA00003429.

2. Any changes to the research must be submitted and approved by the IRB prior to implementation.

3. If any unanticipated problems occur, they must be reported to the IRB office within 5 business days.

Please forward a copy of this letter to your faculty sponsor, if applicable.
Thank you.

Sincerely,

Deb Paxton
NC State IRB

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
GUIDELINES FOR A PROPOSAL NARRATIVE

In your narrative, address each of the topics outlined below. Every application for IRB review must contain a proposal narrative, and failure to follow these directions will result in delays in reviewing/processing the protocol.

A. INTRODUCTION

1. Briefly describe in lay language the purpose of the proposed research and why it is important.

We are conducting research on how to assist developers to fix software defects. Program analysis tools can potentially make a developer's job easier by helping the developer find software defects early in the development process. However, most of those tools do not fit into the current workflows in fixing defects. Through this research, we hope to find a way of improving the usability of static analysis tools so that they fit better into developers' workflows, allowing them to fix software defects in an interactive manner.

2. If student research, indicate whether for a course, thesis, dissertation, or independent research.

This research is Yoonki Song's Doctoral dissertation.

B. SUBJECT POPULATION

1. How many subjects will be involved in the research?
Estimates or ranges are acceptable. Please be aware that if you recruit over 10% more participants than originally requested, you will need to submit a request to modify your recruitment numbers.

There will be between 10 and 20 human subjects involved in our study.

2. Describe how subjects will be recruited. Please provide the IRB with any recruitment materials that will be used.

Subjects are being recruited through undergraduate/graduate students at North Carolina State University and Software engineers in industry.

3. List specific eligibility requirements for subjects (or describe screening procedures), including those criteria that would exclude otherwise acceptable subjects.

To complete this study, subject must:
- Familiar with programming in Java
- Have some experience with Eclipse
- Have basic knowledge about static analysis to analyze code.

4. Explain any sampling procedure that might exclude specific populations.

Since this is a user study for software developers, we would exclude those who have no programming experience. We prefer developers who have used static analysis tools.

5. Disclose any relationship between researcher and subjects - such as, teacher/student; employer/employee.

There are no relationships between the researchers and subjects.

6. Check any vulnerable populations included in study:

- ☐ minors (under age 18) - if so, have you included a line on the consent form for the parent/guardian signature
- ☐ fetuses
- ☐ pregnant women
- ☐ persons with mental, psychiatric or emotional disabilities
- ☐ persons with physical disabilities
- ☐ economically or educationally disadvantaged
- ☐ prisoners
- ☐ elderly
- ☐ students from a class taught by principal investigator
- ☐ other vulnerable population.

7. If any of the above are used, state the necessity for doing so. Please indicate the approximate age range of the minors to be involved.

N/A

C. PROCEDURES TO BE FOLLOWED

1. In lay language, describe completely all procedures to be followed during the course of the experimentation. Provide sufficient detail so that the Committee is able to assess potential risks to human subjects. In order for the IRB to completely understand the experience of the subjects in your project, please provide a detailed outline of everything subjects will experience as a result of participating in your project. Please be specific and include information on all aspects of the research, through subject recruitment and ending when the subject's role in the project is complete. All descriptions should include the informed consent process, interactions between the subjects and the researcher, and any tasks, tests, etc. that involve subjects. If the project involves more than one group of subjects (e.g. teachers and students, employees and supervisors), please make sure to provide descriptions for each subject group.

To begin the study, we will be contacting each subject that we will be involving with a pre-questionnaire for them to fill out and a consent form for them to sign. Once the documents are returned, we will figure out the best day and time to conduct each study based off the subjects' availability.

Once the time arrives to conduct the study, we will conduct each study at our lab (EB II 3222) or via screen sharing.

The study procedure is the following: (1) *First*, we give short introduction and training session to our prototype. To compare the effectiveness and the efficiency of our prototype, we will divide them into two groups. (2) *Then*, we are going to give them 4 programming tasks. (3) *Next*, we will ask them to complete 2 out of 4 tasks using our prototype and 2 out of 4 tasks without using our prototype. We will observe how the participants do to the given programming tasks with/without our prototype. (4) At the end of the programming tasks, we will give post questionnaire to figure out how the participants satisfy on our prototype (User interface and design guidelines?).

Once the study is over, we ask the subject if we may keep their code for further analysis and thank them for their time. We also give them the opportunity to ask any questions they may have.

2. How much time will be required of each subject?

Each subject will need approximately 40 minutes to an hour for the study.

D. POTENTIAL RISKS

1. State the potential risks (psychological, social, physical, financial, legal or other) connected with the proposed procedures and explain the steps taken to minimize these risks.

A potential risk is that the subject feels as if their abilities as a programmer are being challenged, which could cause them to give insufficient feedback. To minimize the risk of this happening, we plan to make sure to let the subject know before the study starts that anything we say or any questions we ask are just to dig deeper, not to offend. Also, we will be careful in how we word the tasks and express them to the subjects.

2. Will there be a request for information that subjects might consider to be personal or sensitive (e.g. private behavior, economic status, sexual issues, religious beliefs, or other matters that if made public might impair their self-esteem or reputation or could reasonably place the subjects at risk of criminal or civil liability)?

No.

a. If yes, please describe and explain the steps taken to minimize these risks.

3. Could any of the study procedures produce stress or anxiety, or be considered offensive, threatening, or degrading? If yes, please describe why they are important and what arrangements have been made for handling an emotional reaction from the subject.

4. How will data be recorded and stored?

The study will be recorded onto a laptop (video and audio) and stored onto the laptop. We will also be taking notes by hand.

a. How will identifiers be used in study notes and other materials?

Each participant will be assigned an alias and will be referred to as such in study notes and other materials.

b. How will reports will be written, in aggregate terms, or will individual responses be described?

Generally, reports will be written in aggregate terms, however where there are specifically poignant or useful responses, individual responses may be described but no responses that could directly identify an individual will be used.

5. If audio or video recordings are collected, will you retain or destroy the recordings? How will recordings be stored during the project and after, as per your destruction/retention plans?

Video/Audio recordings will be destroyed at the conclusion of the study. Notes and transcripts will be stored on a secure, password-protected server.

6. Is there any deception of the human subjects involved in this study? If yes, please describe why it is necessary and describe the debriefing procedures that have been arranged.

No.

E. POTENTIAL BENEFITS

This does not include any form of compensation for participation.

1. What, if any, direct benefit is to be gained by the subject? If no direct benefit is expected, but indirect benefit may be expected (knowledge may be gained that could help others), please explain.

No.

F. COMPENSATION

Please keep in mind that the logistics of providing compensation to your subjects (e.g., if your business office requires names of subjects who received compensation) may compromise anonymity or complicate confidentiality protections. If, while arranging for subject compensation, you must make changes to the anonymity or confidentiality provisions for your research, you must contact the IRB office prior to implementing those changes.

1. Describe compensation

There will be no direct benefits. We will use the information we gather from this study as a basis for the design of tools that would be available to software developers.

2. Explain compensation provisions if the subject withdraws prior to completion of the study.

N/A

3. If class credit will be given, list the amount and alternative ways to earn the same amount of credit.

N/A

G COLLABORATORS

1. If you anticipate that additional investigators (other than those named on **Cover Page**) may be involved in this research, list them here indicating their institution, department and phone number.

Brittany Johnson, Ph.D. Student, NC State University CSC , bijohnso@ncsu.edu

2. Will anyone besides the PI or the research team have access to the data (including completed surveys) from the moment they are collected until they are destroyed.

No, only the research team will have access to the data.

H. CONFLICT OF INTEREST

1. Do you have a significant financial interest or other conflict of interest in the sponsor of this project? *No*

2. Does your current conflicts of interest management plan include this relationship and is it being properly followed? _____

I. ADDITIONAL INFORMATION

1. If a questionnaire, survey or interview instrument is to be used, attach a copy to this proposal.

2. Attach a copy of the informed consent form to this proposal.

3. Please provide any additional materials that may aid the IRB in making its decision.

J. HUMAN SUBJECT ETHICS TRAINING

*Please consider taking the Collaborative Institutional Training Initiative (CITI), a free, comprehensive ethics training program for researchers conducting research with human subjects. Just click on the underlined link.

FixBugs User Study Training Script

Hello, my name is ..... We are currently conducting research on improving the usability of static analysis tools. As part of our research, we are conducting a user study on FixBugs, an extension of the FindBugs Eclipse plug-in that we have developed based on results obtained from a previous study. The key feature of this tool is an interactive quick fix which is meant to improve the "Quick Fix"/"Quick Assist" functionalities offered by Eclipse. A "Quick Fix" is offered when the code is broken and a "Quick Assist" is a minor refactoring offered by Eclipse.

Today, we will be asking you to use and evaluate two different versions of Quick Fix. During this evaluation, you will be asked to complete a set of tasks; each task is comprised of 3 subtasks). In each subtask, you will be asked to fix a defect using a different approach; the first subtask will be done _____, the second subtask will be done _____, and the third subtask will be done _____. Each task is a different level of difficulty; the first being simplest the last being the most complex or difficult. No task should take more than ____ minutes to complete. ** These blank values will be decided on a rolling basis (the time will be decided after pilot study feedback).

During the study, we encourage thinking aloud as you are making your decisions. We will not be asking any questions or prompting any conversation, but we hope you will share with us your thought process as you are completing the tasks.

Before we begin the tasks for the study however, we will walk you through an example to explain how the process goes and any background on FixBugs you will need to know before using it. You will be able to ask minor questions if needed during the study, but in order to get reliable feedback we hope to avoid this by giving a thorough explanation now.

The first thing to understand about the new version of Quick Fix is how to invoke the tool. There are two ways to invoke a "Quick Fix"/"Quick Assist") in order to utilize a quick fix:
1) click the bug marker in the left column 2) use the keyboard shortcut (Ctrl + 1) or 3) put your cursor on the line where the bug is and wait for the pop up.

Ask participant to go ahead and do this.

You can also see that when you click the bug marker, a description (provided by FindBugs) is available on the bottom of the screen. You can use this information during the study as well to help you decide what warning means and what the best fix will be.

Okay, now that the menu is here we have ___ options; one is an interactive "Quick Fix", one is a normal "Quick Fix" and the other is a "Quick Assist". If you click the "Quick Fix" or "Quick Assist", the fix will be immediately applied to the code (maybe prompt the user to click one/both).

Now, if you undo what has been done, we will go back to examine how the interactive "Quick Fix" works.

Have participant undo changes and re-invoke the menu to click the interactive fix.

Now you see a box has popped up and your code has been highlighted yellow. 'Yellow' highlighting represents your original code. In the pop-up box you can see a some radio buttons - these radio buttons represent the options you have. You can see that there is one button labeled 'Original'; this button is used to revert back to your original code.

Ask participant to click one of the fix options.

Now you see some new code has been introduced and there are different colors of highlighting. The fix has been temporarily applied to the code for you to preview before accepting (the option to revert to your original code is always there). Any code that has been changed from the original code is highlighted 'orange' and code that is new is highlighted 'green'.

The interactive "Quick Fix" also allows for rename refactoring if needed. If you click on the variable _____ you can see that there is a light gray box around it and all other instances of the variable. Changing this variable will change all other variables with this box around it. If you move your cursor out of the box, the interactive pop up goes away but if you move the cursor back into the box the pop up comes back. Ask participant to try. In order to get rid of the pop up box (in this case you would have to reinvoke the tool to get your options back), you can either 1) press 'esc' on your keyboard , 2) click the 'X' in the upper left hand corner, or 3) click outside of the pop up box in the editor. Ask participant to try.

Another feature of the interactive "Quick Fix" is the ability to move the "Quick Fix" options box from its default position (right about the first line of the affected code) to any corner of your screen just by dragging and dropping the box.

Once you have selected the fix you approve of, you can click outside of the "Quick Fix" options box to get it and the highlighting to go away. During the study, since it is possible to accidentally click outside the box, we will wait to hear "I'm done" or "I'm satisfied with this solution" to count that task completed.

Prompt the user to 'accept' the fix.

This "Quick Fix" also offers drag and drop functionality for fixing bugs that may require more user input than just selecting a fix. Here we have a simple example of _____. Once you choose to fix the problem, the tool automatically applies a solution which may not necessarily be the way you want your code to look or function. Ask participant to apply the fix.

Now you can see that there are colored boxes around the synchronization keywords. If you click and drag one of the synchronization keywords around the screen you will see that there are designated 'drop zones' where the keyword can be dropped to modify the code. Get the participant to drop the keyword in one of the drop boxes and explain what each drop box would do. Now you can see that the code has been altered based on where you dropped your sync keyword. You also still have the option of going back to the original code.

Now, are there any questions before we complete this training session and move on to the tasks?

Screenshots

Original Code

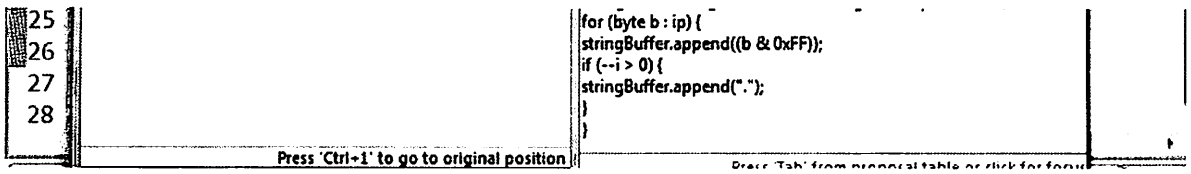
```
 7    public String getAddress() throws Exception {
 8        String result = null;
 9        try {
10            InetAddress address = InetAddress.getLocalHost();
11            byte[] ip = address.getAddress();
12
13            int i = 4;
14            String ipAdddress = "";
15            for (byte b : ip) {
16                ipAdddress += (b & 0xFF);
17                if (--i > 0) {
18                    ipAdddress += ".";
19                }
20            }
21            result = ipAdddress;
22        } catch (UnknownHostException e) {
23            e.printStackTrace();
24        }
25        return result;
26    }
```

"Quick Fix" Version 1

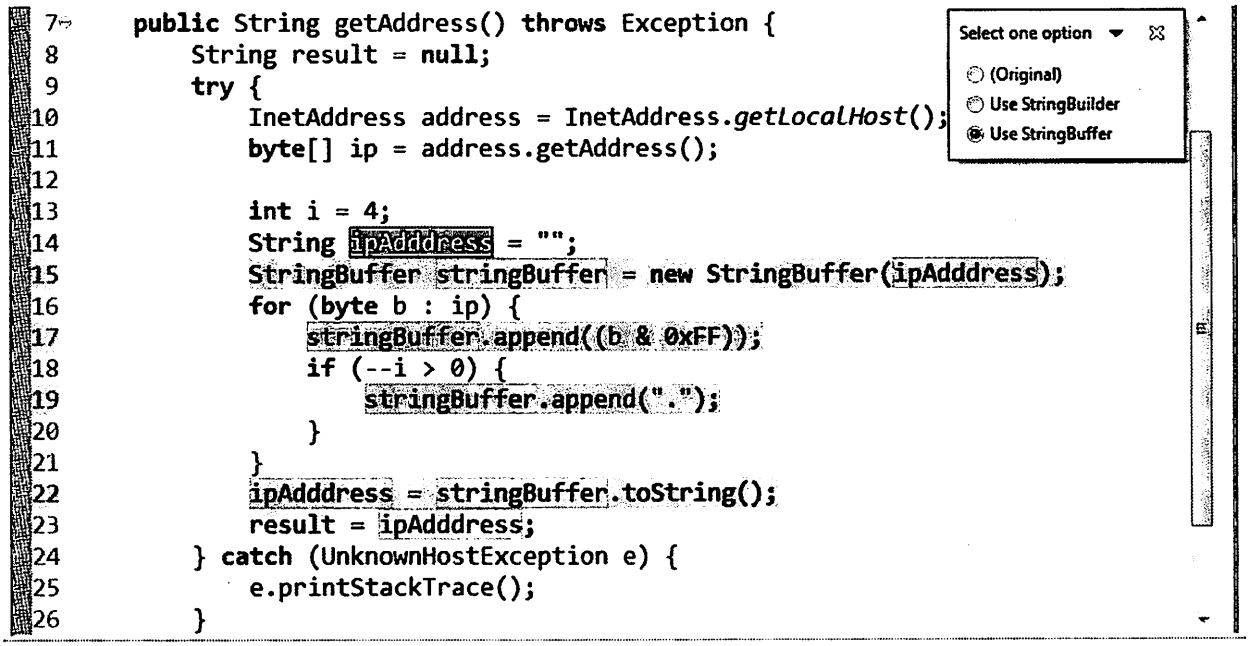
```
 7    public String getAddress() throws Exception {
 8        String result = null;
 9        try {
10            InetAddress address = InetAddress.getLocalHost();
11            byte[] ip = address.getAddress();
12
13            int i = 4;
14            String ipAdddress = "";
15            for (byte b : ip) {
16                ipAdddress += (b & 0xFF);
17                if (--i > 0) {
18                    ipAdddress += ".";
19
20
21
22
23
24
```

[FixBugs] StringBufferConcatenationResolution
[QuickFix] Use StringBuffer
[QuickFix] Use StringBuilder

```
public String getAddress() throws Exception {
String result = null;
try {
InetAddress address = InetAddress.getLocalHost();
byte[] ip = address.getAddress();

int i = 4;
String ipAdddress = "";
StringBuffer stringBuffer = new StringBuffer(ipAdddress);
```

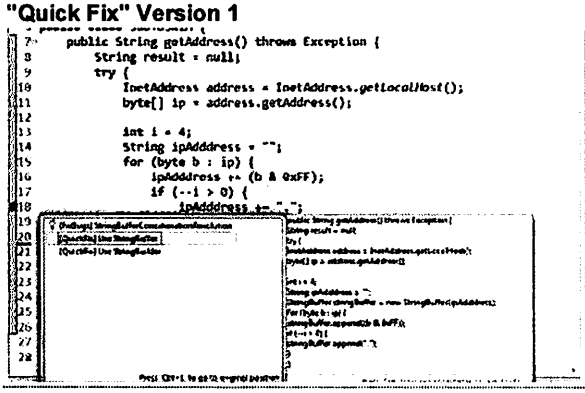


"Quick Fix" Version 2

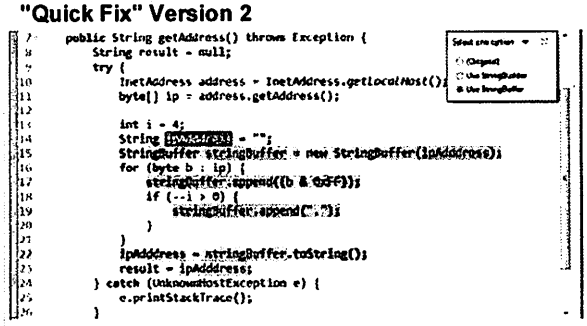


Quick Feedback

"Quick Fix" Version 1



"Quick Fix" Version 2



"Quick Fix" version 1 allows me to more quickly understand what will happen to my code once the fix is applied.

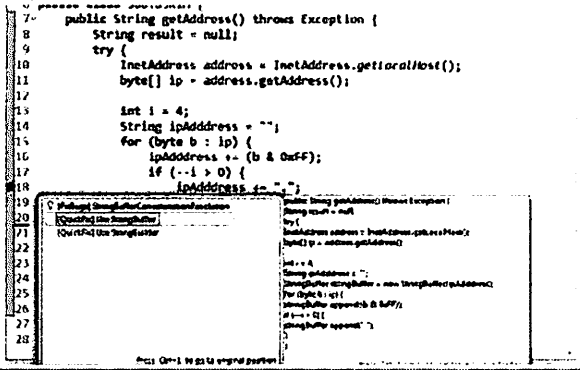
Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

"Quick Fix" version 2 allows me to more quickly understand what will happen to my code once the fix is applied.

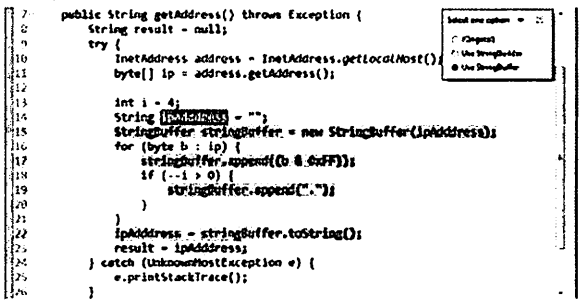
Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

Minimization of user actions

"Quick Fix" Version 1



"Quick Fix" Version 2



"Quick Fix" version 1 reduces the amount of work required to fix a bug.

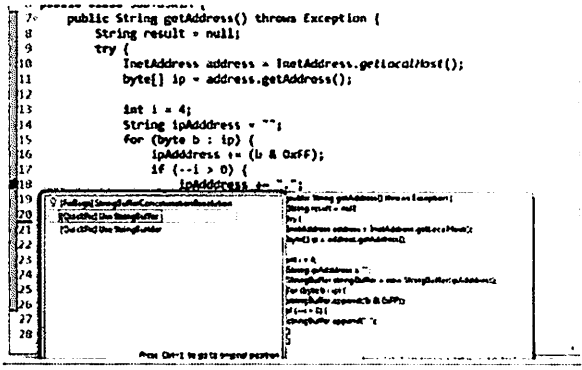
Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

"Quick Fix" version 2 reduces the amount of work required to fix a bug.

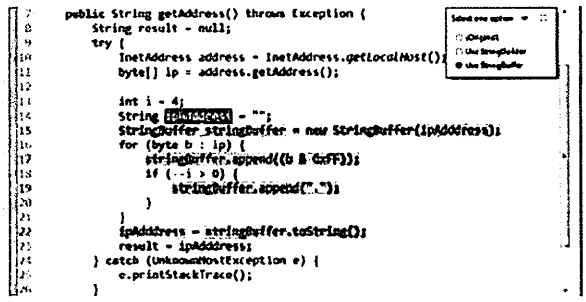
Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

Direct Update

"Quick Fix" Version 1



"Quick Fix" Version 2



"Quick Fix" version 1 provides an efficient and helpful approach to previewing a potential bug fix directly in my code

"Quick Fix" version 1 provides an efficient and helpful approach to previewing a potential bug fix directly in my code.

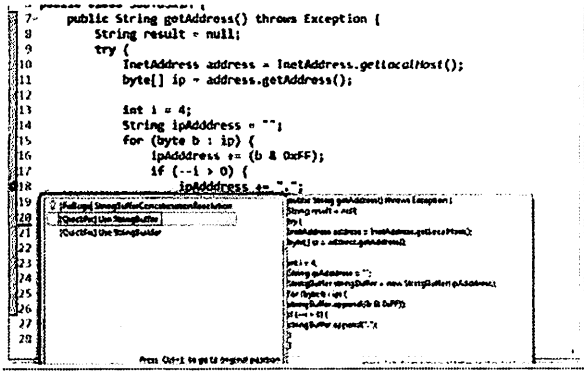
Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

"Quick Fix" version 2 provides an efficient and helpful approach to previewing a potential bug fix directly in my code.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

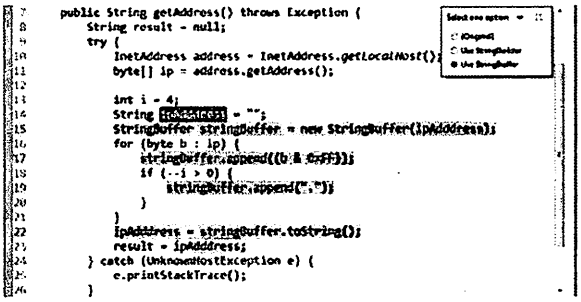
Support for rename refactoring

"Quick Fix" Version 1



```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        for (byte b : ip) {
            ipAddress += (b & 0xFF);
            if (--i > 0) {
                ipAddress += ".";
```

"Quick Fix" Version 2



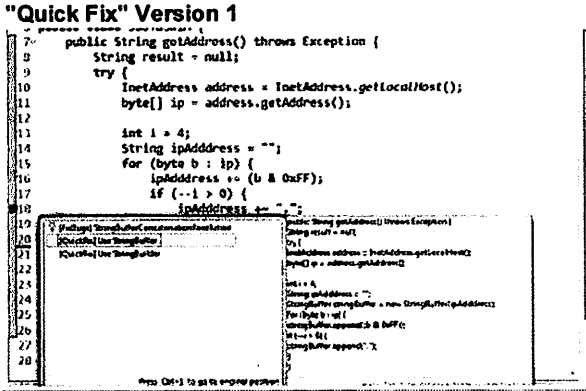
```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        StringBuffer stringBuffer = new StringBuffer(ipAddress);
        for (byte b : ip) {
            stringBuffer.append((b & 0xFF));
            if (--i > 0) {
                stringBuffer.append(".");
            }
        }
        ipAddress = stringBuffer.toString();
        result = ipAddress;
    } catch (UnknownHostException e) {
        e.printStackTrace();
    }
```

I find it useful that "Quick Fix" version 2 automatically invoked the rename tool for me.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

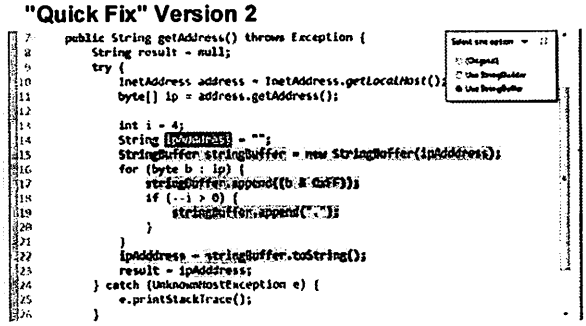
Undoable/Redoable Options

"Quick Fix" Version 1



```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        for (byte b : ip) {
            ipAddress += (b & 0xFF);
            if (--i > 0) {
                ipAddress += ".";
```

"Quick Fix" Version 2



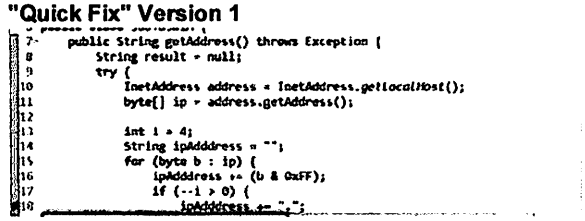
```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        StringBuffer stringBuffer = new StringBuffer(ipAddress);
        for (byte b : ip) {
            stringBuffer.append((b & 0xFF));
            if (--i > 0) {
                stringBuffer.append(".");
            }
        }
        ipAddress = stringBuffer.toString();
        result = ipAddress;
    } catch (UnknownHostException e) {
        e.printStackTrace();
    }
```

"Quick Fix" version 2 makes the process of switching between fixes faster.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

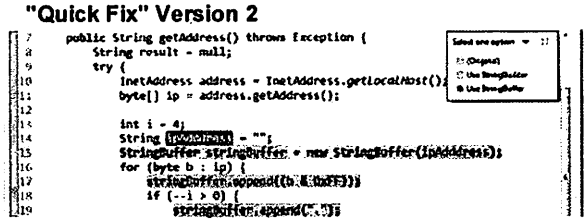
Color annotations

"Quick Fix" Version 1

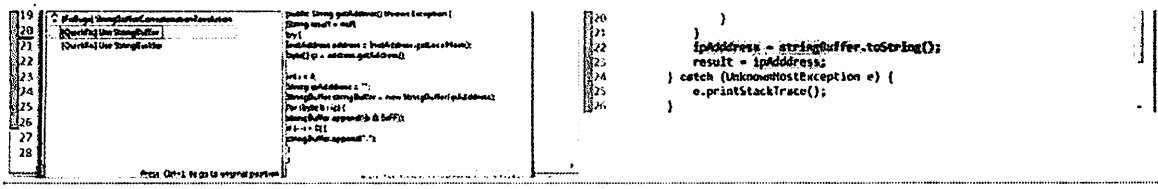


```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        for (byte b : ip) {
            ipAddress += (b & 0xFF);
            if (--i > 0) {
                ipAddress += ".";
```

"Quick Fix" Version 2



```
public String getAddress() throws Exception {
    String result = null;
    try {
        InetAddress address = InetAddress.getLocalHost();
        byte[] ip = address.getAddress();

        int i = 4;
        String ipAddress = "";
        StringBuffer stringBuffer = new StringBuffer(ipAddress);
        for (byte b : ip) {
            stringBuffer.append((b & 0xFF));
            if (--i > 0) {
                stringBuffer.append(".");
```



```
        }
      }
      ipAddress = stringBuffer.toString();
      result = ipAddress;
    } catch (UnknownHostException e) {
      e.printStackTrace();
    }
```

It is useful to be able to differentiate between new, modified and old code when fixing a bug.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

"Quick Fix " version 1 makes it easy to differentiate between original, new and modified code.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

The color annotations used by "Quick Fix" version 2 makes it easy to differentiate between original, new and modified code.

Strongly Disagree	Disagree	Neither Agree nor Disagree	Agree	Strongly Agree
○	○	○	○	○

FixBugs: Consent Form and Pre-Questionnaires

This three page form is used to sign up for a user study on an interactive quick fix static analysis feature (named FixBugs)

To complete this study, you must:
- Be familiar with programming in Java
- Have some experience with Eclipse IDE (www.eclipse.org/)

* Required

By the definition above, I am eligible to participate. *

- ○ Yes
- ○ No

Continue »

Powered by Google Docs

FixBugs: Consent Form and Pre-Questionnaires

* Required

Consent Form (Page 2 of 3)

North Carolina State University
INFORMED CONSENT FORM for RESEARCH
Title of Study: Interactive Bug Fixing

Principal Investigator: Yoonki Song
Other Investigators: Brittany I. Johnson

Faculty Sponsor: Emerson Murphy-Hill

What are some general things you should know about research studies?
You are being asked to take part in a research study. Your participation in this study is voluntary. You have the right to be a part of this study, to choose not to participate or to stop participating at any time without penalty. The purpose of research studies is to gain a better understanding of a certain topic or issue. You are not guaranteed any personal benefits from being in a study. Research studies also may pose risks to those that participate. In this consent form you will find specific details about the research in which you are being asked to participate. If you do not understand something in this form it is your right to ask the researcher for clarification or more information. A copy of this consent form will be provided to you. If at any time you have questions about your participation, do not hesitate to contact the researcher named above.

What is the purpose of this study?
We are conducting research on how to improve the usability of static analysis tools for developers. Static analysis can potentially make a developer's job easier by helping the developer find bugs early in the development process. However, developers are not using static analysis tools as often as the benefits might indicate. The problem may be that current static analysis tools do not fit into the current workflows of developers. Through this research, we hope to find a way of improving static analysis tools so that they fit better into developers' workflows, allowing them to detect and fix bugs in their code faster and earlier in the development process.

What will happen if you take part in the study?
If you agree to participate in this study, you will be asked to complete a set of programming tasks with and without the aid of our static analysis tool prototype (FixBugs). You will undergo a 10-15 minute training session prior to completing the tasks where we will explain the concepts you need to understand to successfully complete the study. Once finished, you will then be asked to evaluate FixBugs using a predefined set of heuristics in the form of a post-study survey. (45 – 60 minutes)

Risks
There are no risks we foresee in conducting this user study.

Benefits
There will be no direct benefits to you for your participation in this study. We will use the information we gather from this study as a basis for the design of tools and activities that would be available to software developers.

Confidentiality
Information we obtain in the study will be kept confidential to the fullest extent allowed by law. Data will be stored securely in a locked laboratory and on password secure servers. We will maintain a confidential spreadsheet that links the real names and email addresses of participants with an assigned pseudo-name that will be used in publications. This spreadsheet will be kept entirely

separately from any comments you make during the study. You will be given the opportunity to allow the researchers to anonymously publish some or all the data.

Compensation
You will not receive anything for participating.

What if you have questions about this study?
If you have questions at any time about the study or the procedures, you may contact the researchers, Yoonki Song, at EB II 3222 / [919/673-9194] or Brittany Johnson, at EB II 3222 / [919/817-8371]

What if you have questions about your rights as a research participant?
If you feel you have not been treated according to the descriptions in this form, or your rights as a participant in research have been violated during the course of this project, you may contact Deb Paxton, Regulatory Compliance Administrator, Box 7514, NCSU Campus (919/515-4514).

May we record the interview (screen and audio only) for post-interview analysis? *

- ◯ Yes
- ◯ No

Informed Consent *

☐ "I have read and understand the above information. I can save a copy of this webpage for my records. I agree to participate in this study with the understanding that I may choose not to participate or to stop participating at any time without penalty or loss of benefits to which I am otherwise entitled."

Your Email Address *

« Back Continue »

FixBugs: Consent Form and Pre-Questionnaires

* Required

Pre-Questionnaires (Page 3 of 3)

Before the study, we'd like to know a few things about you.

Which of the following Integrated Development Environments (IDEs) for Java have you used? *

For more information, please visit
http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments#Java

- [ ] BlueJ
- [ ] Eclipse
- [ ] Greany
- [ ] Greenfoot
- [ ] IntelliJ IDEA
- [ ] JBuilder
- [ ] JCreator
- [ ] JDeveloper
- [ ] KDevelop
- [ ] NetBeans
- [ ] Rational Application Developer
- [ ] Servoy
- [ ] Xcode
- [ ] Other:

How many years of software development experience do you have? *

Please enter a number in years. (e.g 2 or 3.5)

How many years of experience do you have programming in Java? *

Please enter a number in years. (e.g 2 or 3.5)

How many years of Eclipse Experience do you have? *

Please enter a number in years. (e.g 2 or 3.5)

Which of the following static analysis tools have you used? *

For more information, please visit
http://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis#Java

- [ ] CheckStyle
- [ ] FindBugs
- [ ] Hammurapi
- [ ] PMD
- [ ] Soot
- [ ] Squale
- [ ] Jtest
- [ ] LDRA Testbed
- [ ] SemmleCode
- [ ] SonarJ
- [ ] Kalistick
- [ ] Other:

Have you ever used the "Quick fix" or "Quick Assist" feature offered by Eclipse? *
For more information, please visit http://wiki.eclipse.org/Image:Quickfix1.jpg

- ◯ Yes
- ◯ No

« Back Continue »

FixBugs: Consent Form and Pre-Questionnaires

Please Press the Submit Button Below

If you are satisfied with your answers, please use the submit button below.

« Back Submit