Assignment 1

Patrik Ackland

- **Understanding a large codebase.** I find that it is difficult to navigate large codebases to find out where new functionality should be added or what section of the code a certain bug report referring to. This problem usually occures when you are new to a codebase. As you get more familiar with the code this problem disappears. However, when additional people are added to the project they will encounter the same problem. Even if you know parts of the code you might want to get familiar with another part of the code which introduces this problem again. A large number of files is not the only problem. Design documents that are supposed to give this information might be poorly written or out of date. This could be addressed by automatically generating design documents from the cosebase. The data that can be used for this is the code itself and discussions between developers. Perhaps the original design documents can be used to generate the design and the version control system can be used to update the documents automatically. There are tools that do this to a certain extent. Microsoft Visual Studio can generate diagrams that show how classes interact with each other. However this usually limited in helping out with a large codebase.

- **Writing good documentation.** I find it difficult to write useful documentation that help other people understand my code. The underlying cause is that I often feel that I am explaining my work twice. Once with code and once with documentation. This leads to not putting as much effort into documentation. Another reason for this is that documentation is usually an afterthought. First I write the code and then I come back to write the documentation. Many of the tools that exist to help with documentation such as autogenerating javadoc in Eclipse only generates annotations for parameters and their names but does not assist in writing the actual documentation. Tools that autogenerate documentation are a way to solve this problem. They could use the code and design documents as data. It might be hard to generate good documentation but it takes less effort from the programmer to correct a few mistakes in otherwise good documentation.

- **Writing useful testcases.** I find that it is often difficult to write test cases that cover every possible case. I often write tests that cover the normal use cases that would get run most of the time. However, the edge

Summary of Comments on ackland.pdf

Page: 1

Number: 1 Author: mernst   Subject: Highlight   Date: 1/11/2016 9:38:06 PM
Use a more specific title. What is the document about?

Number: 2 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:33:09 AM
Is it that the bug report is referring to the code, or you talking about the defective code that caused the fault?

Number: 3 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:35:13 AM
Format and content of these documents? Exactly what information they contain, and how to be organized? In what way would that help programmers to hone in on the part of the code that is relevant to them?

Be specific about the problem and the solution. You've mentioned a very broad and general problem and given only a vague inkling of its specifics or how it might be solved. This is unlikely to be a successful way of selecting a high impact, interesting project.

Number: 4 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:36:05 AM
Why? What is this information good for, and what is it not good for? What alternate or different information is needed?

Number: 5 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:36:36 AM
Can you define "useful"? What sort of documentation are you looking for?

Number: 6 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:37:02 AM
There also the tests that are another way of explaining your code, so the problem they be even worse than this.

Number: 7 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:38:47 AM
Is this a good methodology? That is is more effective overall even though it might have some negatives? Where is it a negative and undesirable methodology?

Number: 8 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:39:55 AM
Do you think it is reasonable to assume that this exist? If we are talking about a programmer and methodology where the documentation does not get written, can we assume that there is extensive design documentation to work from? Extensive

cases that are forgotten about might be the ones that end up [1]causing the most trouble later. The most common use cases would [2]cause a bug report quickly. [3]Ways to currently solve this [4]problem is to use profiling techniques to [5]see what parts of the code is being executed. Another way is to [6]randomize input. One way this could be solved is to automatically generate test cases from the code itself and perhaps use other data as a guide to what inputs should be used[7] I think the [8]reason this has not been solved effectivly is because [9] is hard to reason about what test cases are needed. The automatically generated test cases need to do [10]better job than a human would.

- **Lack of effective autocomplete.** When writing code I feel that more effective autocomplete would be useful. Current autocomplete tools can assist with filling in names of objects and functions or give the code for a general for loop. However, I think a lot more can be done to reason about the program that is being written and what code the programmer is expecting to write. If I write a function with the word *sum* in the function name which takes an array as an argument, it could be inferred that I would like the sum of the elements in the array [11] a result of the function. This kind of autocomplete would assist the programmer with simple tasks and let the programmer focus on more difficult tasks that cannot be generated automatically. This can be extended in other functions to using recently used or recently declared variables[12]The data for this kind of analysis could be previously written code. With machine learning techniques it could start giving poor suggestions and learn more as it gets used to the programmers [13]de. Many IDEs can automatically generate functions with [14]neralized names and arguments based on a call to a function that does not exist. These are often [15]neral and do not help with the actual code. I find it easier to write the function declaration myself since Eclipse also generates comments about the function [16]ing autogenerated that I have to remove.

Number: 1 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:41:45 AM

Number: 2 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:41:35 AM

I'm not sure what it means to cause a bug report; a bug report is something initiated by user.

I'm also not sure what you mean that the use case would cause a bug report. I would think that a fault, or defective code, would lead to a bug report.

Number: 3 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:42:21 AM

Are you claiming that this is the way that programmers actually perform this task today? I'm not interested as much in potential research ideas as in the practicalities and limitations of current practice.

Number: 4 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:42:21 AM

Number: 5 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:45:15 AM

This does not create a test.

Number: 6 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:44:56 AM

Randomizing input does not create a test case; in the best case, it creates inputs, which are not complete tests.

Number: 7 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:49:52 AM

The suggestion is vague and nonspecific; it doesn't give me any idea what you're thinking (if you have specifics in mind) nor how to go about implementing it, nor how to give feedback on the idea. You've mentioned some interesting topics in this document, but I don't feel there are concrete ideas in it. So this is a start, but you need to get specific. Just

Number: 8 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:48:24 AM

Number: 9 Author: mernst   Subject: Highlight   Date: 1/12/2016 5:50:12 AM

Was this hard? Be specific.

Number: 10   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:50:30 AM

In what respect? What limitations exist today?

Number: 11   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:56:34 AM

Why not just suggest calling an existing function in this case?

Under what circumstances is it desirable to reimplement and under what circumstances is it desirable to reuse an existing implementation? In cases where a new implantation as needed, a limitation may be that it is less likely that a current implementation exists.

That said, this seems like the most concrete and also most promising of all the proposals in this document. It may be desirable to write a function when no appropriate implementation exists in a library. Nonetheless, many other people may have written implementations in their own code, so searching client code rather than just libraries may lead to code snippets that could be copied. Another example of where this kind of templating or code structure suggestion would be useful is when the implementation needs to be customized. For example, suppose that you want to take the sum of a particular field of your data structure; that is, it's not an array of integers but an array of nodes, where each node has an int field. Automatically guessing the loop that would add the nodes would be useful, even though that code as written would be type incorrect and not exactly what the user wanted as the final code.

Number: 12   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:57:00 AM

I don't follow this part. Can you be more specific?

Number: 13   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:57:35 AM

What aspect of style you thinking of? So far this point has only talked about going from names to code.

Number: 14   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:57:54 AM

In what sense is the name generalized?

Number: 15   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:58:15 AM

In what way are they general? What don't they help with, and why?

Number: 16   Author: mernst   Subject: Highlight   Date: 1/12/2016 5:58:38 AM

Is it not possible to customize the template?