# September

## C instead of Java

### Why? PMD etc

## Splint

## FramaC, Blast, clang

## Eclipse

# October

## 10/9

### Thesis Proposal

## 10/14

### Good progress and proposal. Next steps are finding a parser and consequently a language. Look into Splint and GCC only enough to ensure that building a new setup is the right idea. After language and parser and determined, spend some time both hacking out a “solution” without any form of plug-in/customization. This can then be refactored and referenced while looking into how to build up an architecture. This week also focus on getting documentation/recording setup.

#### Aux: look into getting svn put up on penguins

#### Probable parsers = JavaCC and Bison

### Email: 10/15 - The parser that I used for my dissertation is named Sable, not Sabre. No wonder I couldn't find it yesterday!  Here a reference: <http://sablecc.org/> It seems that a C99 (but not a C90) grammar is available for Sable.  See this reference:  <http://sablecc.sourceforge.net/grammars.html>.  But it also seems that the grammar works only with one particular version of Sable.  Of course that's not good.  So it might be wise to avoid Sable.

### Completed a very rough draft of the Problem Description section.

### IDIOT: Checkstyle and PMD are both customizable!

### Svn repository (off penguins) working with automatic login

## 10/21 as of 10/27

### Ok that other customizable things exist. Makes the architecture possibly a little less hard/challenging/interesting but still filling a void. Still need to focus on grammar/language (probably JavaCC or Bison/Yacc but grammar for JavaCC is a little iffy). Write follow up email at beginning of break but no meeting next week.

### David Walker: In general, it is just fine to write an analyzer for language X in a different language Y.  So you shouldn't be worried about using a Java parser generator to analyze C.  If I were you I would choose to choose JavaCC if Java is the language you are most comfortable programming in.  Also, I recommend making your choice of parser generator based on the fact that someone else had already written a grammar for C using that parser generator.  Writing a parser for C is a hard thing to do -- you don't want to do that yourself.  You want to reuse a parser someone else has written.