

Contemporary Peer Code Review Practices

Jeffrey C. Carver
Nasir U. Eisty

University of Alabama



Hello
my name is

Can you spot the mistakes?



```
1 // file: IVR.CPP
2 void IVR()
3 {
4     //press 1 for account balance, 2 for last transaction,
5     //3 for last statement, any other for operator
6     play_prompt();
7
8     int key_pressed= get_user_choice();
9     if(key_pressed ==1)
10    {
11        play_account_balance();
12    }
13    else if(key_pressed ==2)
14    {
15        play_last_transaction();
16    }
17    else if(key_pressed ==3)
18    {
19        play_last_statement();
20    }
21    else transfer_to_operator();
22 }
```

Assignment operator
instead of comparison

```
1 //file: printer.java
2 if (user.isAuthenticated)
3 {
4     userAccess = checkUserAuthorization(user);
5
6     //if user has access to printer
7     if(user.isAuthenticated && userAccess.printer)
8         printUsageReport ();
9     else
10        emailUsageReport ();
11 }
```

Redundant check

```
1 //file: UserStats.java
2 String[] listOfUsers = getUsersList();
3 //print all the users name
4 for(int i=1;i< listOfUsers.length();i++)
5     System.out.println("User# "+i+": "+listOfUsers[i]);
6
```

Will not print first user

Do you think only novice
developers make those
mistakes?

```
#define CDMA_SUBSCRIPTION_SOURCE_NV 0  
#define CDMA_SUBSCRIPTION_SOURCE_RUIM 1
```

80
81
82

David Turner

Sep 13, 2010

the functions below operate on an AModem object, so should be named amodem_switch_technology().

amodem_set_cdma_xxxx

[Reply ...](#) [Reply 'Done'](#)

Jaime A Lopez-Sollano

Sep 23, 2010

Done

[Reply ...](#) [Reply 'Done'](#)

```
static const char* switchTechnology(AModem modem, AModemTech newtech, int32_t newpreferred);  
static int set_cdma_subscription_source( AModem modem, ACdmaSubscriptionSource ss);  
static int set_cdma_prl_version( AModem modem, int prlVersion);
```

83
84
85
86

```
if (noMedia) {  
    // In case the file is known and now is under a  
    // .nomedia folder mark as not seen in order to  
    // be removed from files table in the post scan.  
    if (entry != null && noMedia) {
```

478
479
480
481
482

Emilio López

Jan 15, 2012

Isn't '&& noMedia' redundant? noMedia can only be true inside the if block.

[Reply ...](#) [Reply 'Done'](#)

David Marques

Jan 16, 2012

Thanks Emilio! My bad :D

[Reply ...](#) [Reply 'Done'](#)

```
        entry.mSeenInFileSystem = false;  
    }  
    return null;  
}
```

483
484
485
486
487

#13244

```
if ((nbytes=recvfrom(fd,msgbuf,MSGBUFSIZE,0,  
                     >>                                (struct sockaddr *) &addr,&  
                     perror("recvfrom"));  
                     exit(1);  
                 }  
                puts(msgbuf);
```

~~Review Status~~

Feb 8, 2010

Isn't there a risk of msgbuf not being terminated? Reading all the way to MSGBUFSIZE seems to risk overwriting the implicit init done, or does the kernel always null-terminate the buffer, even if it is larger than MSGBUFSIZE?

I guess the worst case scenario is puts going off on a tangent, so probably not particularly dangerous, but still?

[Reply](#)

[Done](#)

~~Review Status~~

Feb 9, 2010

You are absolutely right. I will change it to write() the number of bytes received. I only tested it with the other file (sender.c) which sends "Hello, world" with the null-byte attached. I will send a patch when I get home. Thanks for the review!

```
54 »      »      if ( $this->isPermalink ) {  
55 »      »      »      »      $html .= Linker::link(  
56 »      »      »      »      »      SpecialPage::getTitleFor( 'ArticleFeedbackv5', $record[0]->  
age_title ),  
57 »      »      »      »      »      wfMessage( 'articlefeedbackv5-special-goback' )->text());
```

Catrophe

Apr 17, 2012

And this unfixes a security bug that was fixed earlier: this must be `->escaped()`, not `->text()`, because the second param to `link()` is `HTML`.

[Reply ...](#) [Reply 'Done'](#)

Reha Sterbin

Apr 17, 2012

Done

[Reply ...](#) [Reply 'Done'](#)

We all make mistakes and need help identifying them



You might be an Internet 'solopreneur', a lone wolf who writes, edits, formats, checks and publishes your own content. Or you might work for a ~~company~~ website, either as a content creator or as a ~~production~~ editor, sub-editor or editor - the person in charge of quality control and tasked to edit/proofread ~~work~~ produced by your colleagues.

Not you? Maybe you want to self-publish a book on Amazon, check through a student essay or thesis, perhaps error-check a company report? Maybe you're involved in creating a brochure, newsletters, business cards, product packaging or signs? Not you either? What about sending an email, posting a Tweet or updating Facebook? They all involve words. Is everything you've typed recently correct? Are you certain? It's not always easy to spot errors.

You might not have a proofreading process. Or you might be already be actively proofreading and want to get better at it.

If you publish or print words of any kind, you won't want any mistakes to slip through. You want to ensure that what you produce is the best that it can be. This is especially true if you are printing anything. Mistakes on the web can be corrected. You don't have that luxury with print. Print has permanency. Just think how you'd feel if you'd been the sign writer who painted 'SHCOOL' in big white letters on the road when you meant to spell 'SCHOOL'. Or that you'd given the OK to a sign that read 'DRIVE-THRU ENTERANCE'. These are real examples. You'll find more proofreading howlers like these later in this book.

Again, proofreading is an essential part of the publishing process. The larger the font, the more you should check it. The more permanent the publication or installation, the more you should check it. You should check and re-check everything you plan to publish until your eyes ache or until the words start to look incorrect and you have to look them up again just to make sure that you're not going crazy. And if you have time for a re-re-check, so much the better.



Where does it fit?

Code Review

Other Activities



Code Review Goals

Team building



Improved code

```
43 <body ><?php body _c...>
44 <div id="fb-root"></div>
45 <script>(function(d, s, id) {
46   var js, fjs = d.getElementsByTagName(s)[0];
47   if (d.getElementById(id)) return;
48   js = d.createElement(s); js.id = id;
49   js.src = "//connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.66appId=2904133313131313";
50   (document, 'script', 'facebook-jssdk'));</script>
51 <div id="page" class="site">
52   <a class="skip-link screen-reader-text" href="#content"><?php esc_html_e( 'Skip to content', 'urdutube' ) ; ?></a>
53   <header id="masthead" class="site-header" role="banner">
54     <div class="site-branding">
55       <div class="navBtn pull-left">
56         <?php if(is_home() && $xpanel['homepage-style'] == 1) { ?>
57           <i class="fa fa-bars fa-3x"></i></a>
58         <a href="#" id="openMenu"><i class="fa fa-bars fa-3x"></i>
59         <?php } else { ?>
60           <a href="#" id="openMenu2"><i class="fa fa-bars fa-3x"></i>
61         <?php } ?>
62       </div>
63       <div class="logo pull-left">
64         <a href=<?php echo esc_url( home_url() ) ; ?>
65           <img src=<?php echo $xpanel['l...>
66         </a>
67       </div>
68       <div class="search-box hidden-xs hidden-sm">
69         <?php get_search_form(); ?>
70       </div>
71       <div class="submit-btn hidden-xs hidden-sm">
72         <a href=<?php echo get_page_link( ...>
73       </div>
74       <div class="user-info pull-right mr-10">
75         <?php
76           if ( is_user_logged_in() ) {
```



Personal growth

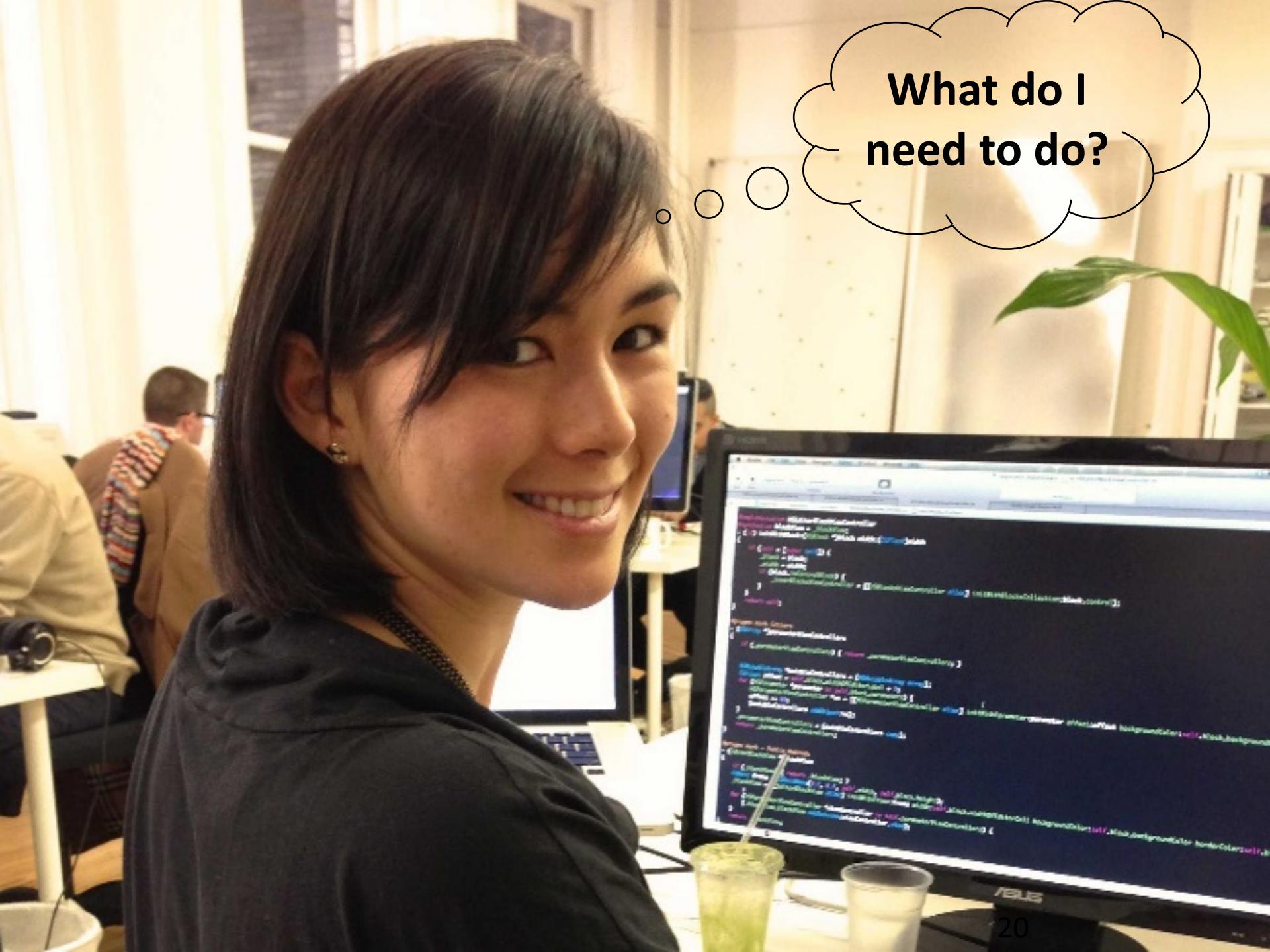
Code Review Goals

- Team building
 - Better shared understanding
 - Team cohesion
 - Peer impression
- Code Quality
 - Find/fix defects early
 - Identify common problems
 - Different perspectives
 - Consistency in code/design
 - More maintainable code
- Personal
 - Learning

A photograph showing a series of small, lit candles arranged in a grid pattern on a dark, reflective surface. The candles are in clear glass holders and are illuminated from within, casting a warm glow. The background is dark, making the light from the candles stand out.

Goal: Achieve peace and harmony

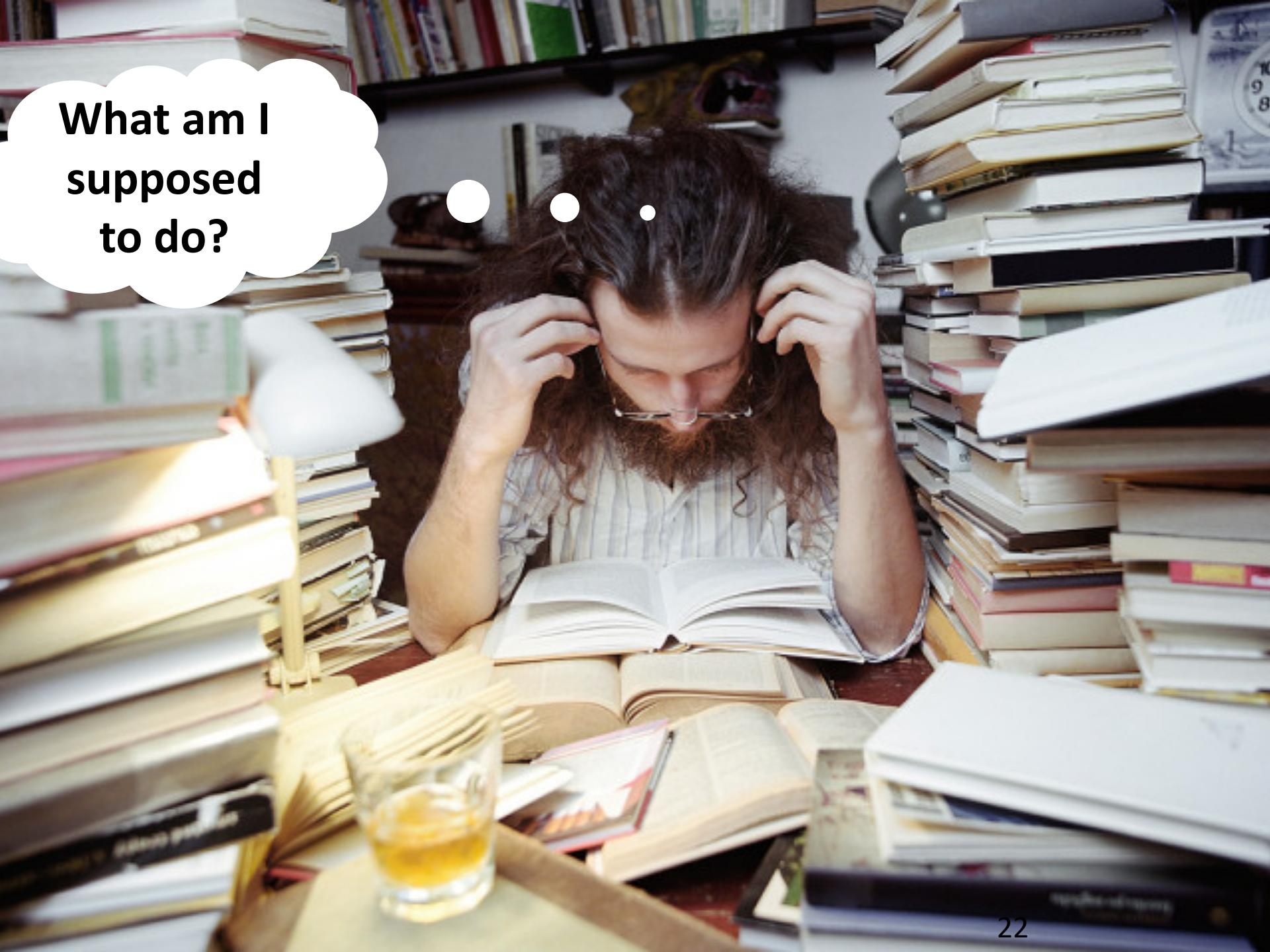
Code Review Practices



What do I
need to do?

For Developers

- Realize that the goal of code review is to improve the overall code, not to evaluate the quality or worth of the developer
- Remove the fear of making mistakes and create an atmosphere where admitting and fixing is OK
- You are not your code
- Be humble
 - You will make mistakes, we all do
 - Someone else will always know more, it's ok, learn from them
 - People bring different perspectives, that's a good thing
- Fight for what you believe, but gracefully accept defeat



What am I
supposed
to do?

For Reviewers

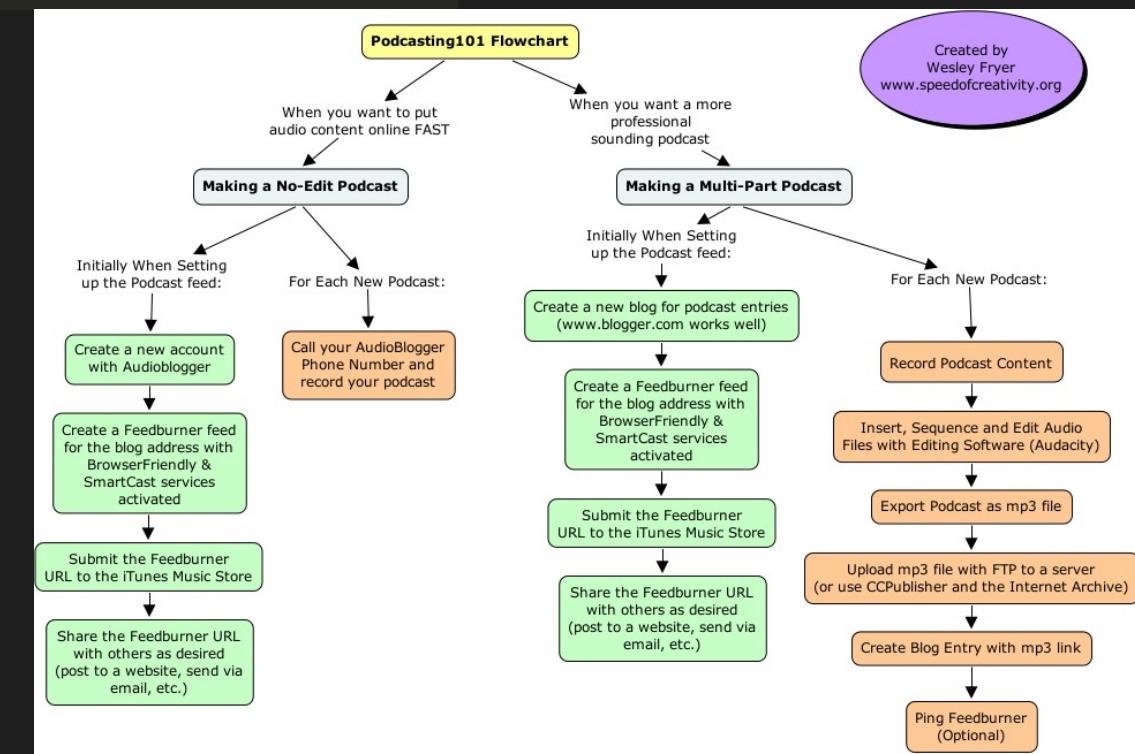
- Focus on the code not the author
 - Use “I” statements rather than “you” statements
 - Criticize the author’s behavior, not their attributes
 - Talk about the code, not the coder
- Ask questions rather than make statements – avoid “why” questions
- Accept that there are different solutions
- Choose carefully which battles to fight
- Remember to praise good code
- Take your time and do it well

Code Review Techniques

Algorithms

```
43 <body><?php body_root() ?>
44 <div id="fb-root"></div>
45 <script>(function(d, s, id) {
46   var js, fjs = d.getElementsByTagName(s)[0];
47   if (d.getElementById(id)) return;
48   js = d.createElement(s); js.id = id;
49   js.src = "//connect.facebook.net/en_US/sdk.js#xfbml=1&version=v2.66appId=200000000000000";
50   fjs.parentNode.insertBefore(js, fjs);
51 }(<document>, 'script', 'facebook-jssdk'));</script>
52 <div id="page" class="site">
53   <a class="skip-link screen-reader-text" href="#content"><?php esc_html_e( 'Skip to content', 'urdutube' ) ?></a>
54   <header id="masthead" class="site-header" role="banner">
55     <div class="site-branding">
56       <div class="navBtn pull-left">
57         <?php if(is_home()) && $xpanel['homepage-style'] == 1) { ?>
58           <a href="#" id="openMenu"><i class="fa fa-bars fa-3x"></i></a>
59         <?php } else { ?>
60           <a href="#" id="openMenu2"><i class="fa fa-bars fa-3x"></i></a>
61         <?php } ?>
62       </div>
63       <div class="logo pull-left">
64         <a href=<?php echo esc_url( home_url() ) ?>>
65           <img src=<?php echo $xpanel['logo']['url'] ?>>
66         </a>
67       </div>
68       <div class="search-box hidden-xs hidden-sm pull-left ml-10">
69         <?php get_search_form(); ?>
70       </div>
71       <div class="submit-btn hidden-xs hidden-sm pull-left ml-10">
72         <?php echo get_page_link($xpanel['submit-link']) ?>> class="header-submit-btn"><i class="fa fa-search fa-3x"></i></a>
73       </div>
74       <div class="user-info pull-right mr-10">
75         <?php
76           if ( is_user_logged_in() ) {
```

Code



Code Review Techniques

- Examine the code
 - Is the code readable to a human?
 - Are variables and method names clear?
 - Is there sufficient documentation for someone to come back 6 months later (or someone new) to understand what the code is doing?
- Examine the algorithms in detail
 - Are there any hidden assumptions, not specified, that could cause problems?
 - Are there edge cases that may not work?
 - What happens with bad or missing data?
 - Does the algorithm do what it is supposed to? – Use stepwise abstraction

Example - Stepwise Abstraction

- Examine the algorithm embedded in the code
- Start at the bottom, extract low-level functionality
- Group low-level functionality into higher-level
- At top level, compare with desired plan

```

while ((a>b) || (b>c) || (c>d))
{
    if(b>a)
    {
        i = b;
        b = a;
        a = i;
    }
    if(c>b)
    {
        i = c;
        c = b;
        b = i;
    }
    if(d>c)
    {
        i = d;
        d = c;
        c = i;
    }
}

```

Assign “b” to “i”
 Assign “a” to “b”
 Assign “i” to “a”

Replace “a” and “b”

Assign “c” to “i”
 Assign “b” to “c”
 Assign “i” to “b”

Replace “b” and “c”

Assign “d” to “i”
 Assign “c” to “d”
 Assign “i” to “c”

Replace “c” and “d”

As long as “a”, “b”, “c”, and “d” are not in descending order

Rearrange “a” and “b” in descending order

Rearrange “b” and “c” in descending order

Rearrange “c” and “d” in descending order

Requirement: Rearrange “a”, “b”, “c”, and “d” in descending order

Code Review Comment Exercise

“You are writing cryptic code”

**“It's hard for me to grasp what is
going on in the code”**

Use I-Messages

**“This is not how I would have solved
the problem”**

**“Why did you use this approach rather
than approach X?”**

Ask questions where possible

“You are sloppy when it comes to writing tests”

“I believe that you should pay more attention to writing tests”

**Criticize the author’s behavior,
not the author**

**“You’re requesting the service
multiple times, which is inefficient”**

**“This code is requesting the service
multiple times, which is inefficient”**

Talk about the code, not the coder

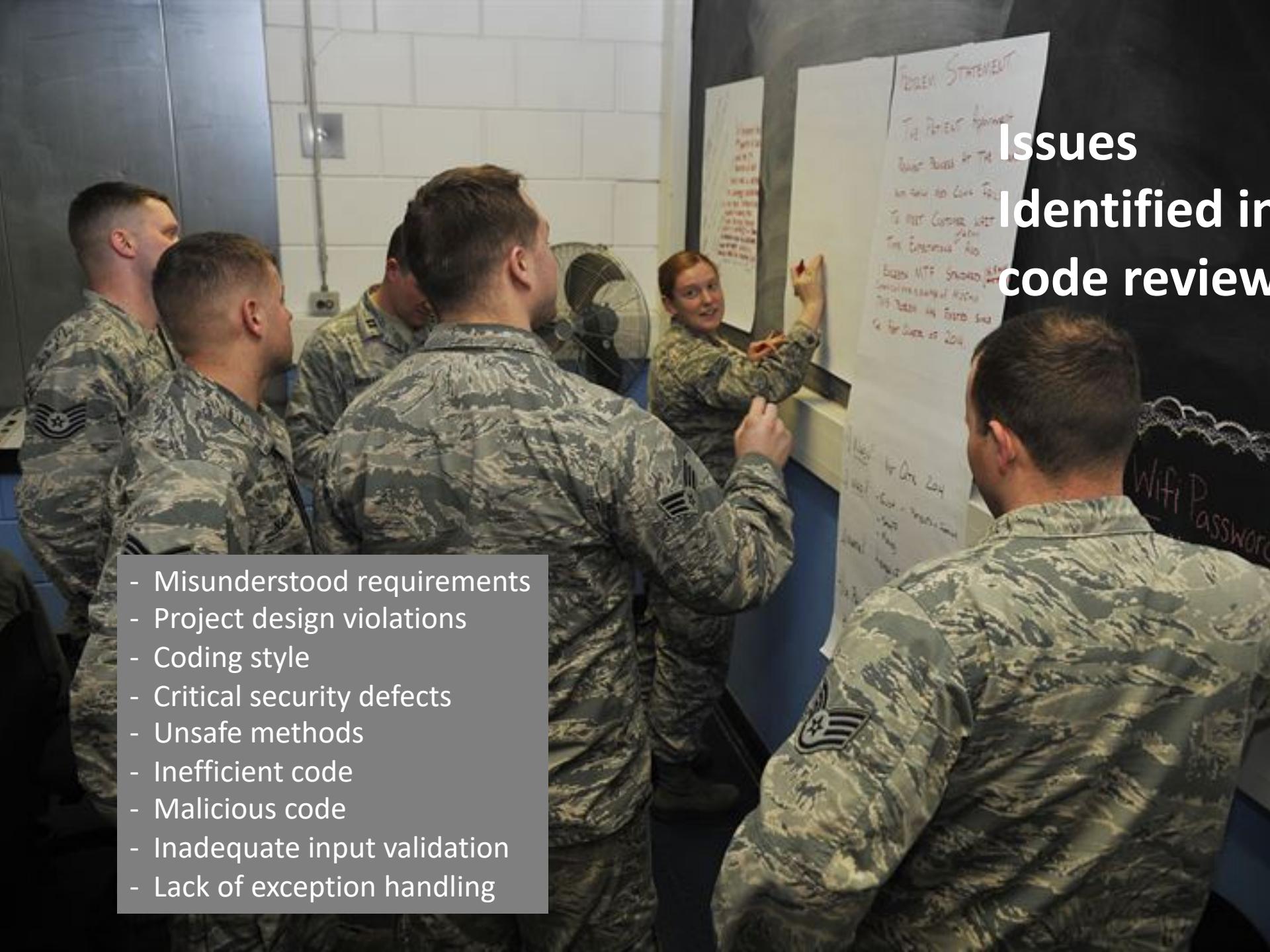
“I always use fixed timestamps in tests and you should too”

“I would always use fixed timestamps in tests for better reproducibility, but in this simple test, using the current timestamp is also ok”

Accept different solutions

Issues Identified in code review

- Misunderstood requirements
- Project design violations
- Coding style
- Critical security defects
- Unsafe methods
- Inefficient code
- Malicious code
- Inadequate input validation
- Lack of exception handling



Developer

checklist

- My code compiles
- My code has been tested and has unit tests
- My code includes appropriate comments
- My code is tidy / follows coding standard
- I have documented corner cases
- I have documented workarounds
- ...
-

Reviewer

checklist

- Comments are understandable and appropriate
- Comments are neither too many or too few
- Exceptions are appropriately handled
- Repetitive code has been factored out
- Frameworks have been used appropriately
- Functionality fits the design/architecture
- Code is testable
- Code compiles

Code Review

BEST PRACTICE



Best Practice

- Practice lightweight code reviews.
- Review fewer than 400 lines of code at a time.
- Inspection rate should be under 500 LOC per hour.
- Do not review for more than 60 minutes at a time.
- Set goals and capture metrics.
- Authors should annotate source code before review.
- Use checklists.
- Establish a process for fixing defects found.
- Foster a positive code review culture.
- Embrace the subconscious implications of peer review.

Scientific Code Review



Reasons to Use Code Review for Scientific/Research Software

- Cultural difference between scientific community and software engineering community
- Correct results are unknown in many cases
- Testing is extensively complex in scientific software
- Common testing approaches may not fit
- May be better to review the scientific algorithm than to extensively test code
- Lack of proper testing knowledge
- Test to check the science, not the software
- Tend to test when development is about to finish

Our Results



Our Results: Findings from Interviews and Surveys

- Positives
 - Large portion of code is reviewed
 - Shared expertise improves code quality
 - Consistent style and reusability
 - Good for new contributors and tricky features
 - Saves debugging time
 - Underlying science is more important than the code
- Challenges
 - Developers are attached to the way they have done things and resist change
 - Lack of time and qualified contributors
 - Lack of enough people to properly review
 - Obtaining reviewer agreement

Our Results: Direct Interactions

- Working directly with a code team for 2-3 months
- Taught them code review
- Anecdotal results
 - Overall positive experience
 - Found faults they would not have tested for
 - Decided to implement a code standard
 - Continued practice

Typical Code Review Workflow

Requests Review



Waitlist

Reviews

Merge

Abandon

PAGE	111	SHEP	APPLE DOS
<hr/>			
3573			
3574	3755	ADF137	
3575	3758	BDE337	
3576	3758	38	
3577	3758	ADE737	
3578	375F	EDE337	
3579	3762	BDE037	
3580	3765	BDE237	
3581			
3582	3768	A990	
3583	376A	BDEC37	
3584	376D	BDE037	
3585	3770	BDF037	
3586			
3587	3773	ADE737	
3588	3776	BDF137	
3589	3779	BDFE36	
3590			
3591	377C	A90A	
3592	377E	BDE137	
3593	3781	0A ^{A945}	
3594	3782	0A ^{EA}	
3595	3783	0A ^{EA}	
3596	3784	8DF36	
3597			
3598	3787	A902	
3599	3789	BDF437	
3600			
3601	378C	209F37	
3602			
3603	378F	ADE337	
3604	3792	BDF137	
3605			
3606	3795	ADE037	
3607	3798	BDE137	
3608	379B	209F37	
3609			
3610	379E	60	
3611			
<hr/>			



Mailing List Code Review

From [REDACTED]
Subject [patch] Fix cross-user symlink race condition vulnerability
Date Wed, 31 Oct 2012 04:46:47 GMT

There is a race condition vulnerability in httpd 2.2.23 (also present in previous releases) that allows a malicious user to serve arbitrary files from nearly anywhere on a server that isn't protected by strict os level permissions. In a shared hosting environment, this is a big vulnerability.

If you would like more information on the exploit itself, please let me know. I have a proof of concept that is able to hit the exploit with 100% success.

This is my first patch submitted to Apache, so I'm sorry if I've missed something. I'm aware that this doesn't meet some of the code standards that are in place (e.g, it doesn't work at all on Windows), but I wanted to put it out there anyway.

The patch that fixes the vulnerability is attached. Thank you in advance for the feedback.

--

[REDACTED]
[REDACTED]
[REDACTED]

Mime • Unnamed multipart/mixed (inline, None, 0 bytes)
 ◦ [Unnamed text/plain](#) (inline, 7-Bit, 857 bytes)
 ◦ [httpd-2.2.23-symlink-protection.patch](#) ([text/x-patch](#)) (attachment, 7-Bit, 8324 bytes)

[View raw message](#)

Pull Requests

Fix a small bug in a project in GitHub Pull Requests

```
#git clone https://github.com/project/code ; cd code
```

```
#vi some.c  
#git commit -a -m 'Fix the frobinator'
```

```
#go to web UI  
#click fork
```

```
#git remote add me https://github.com/$USER/code  
#git push me master
```

```
#go to web UI  
#create pull request
```

Fix a small bug in a project in Gerrithub

```
#git clone https://gerrit.googlsource.com/gerrit ; cd gerrit
```

```
#vi .buckconfig
```

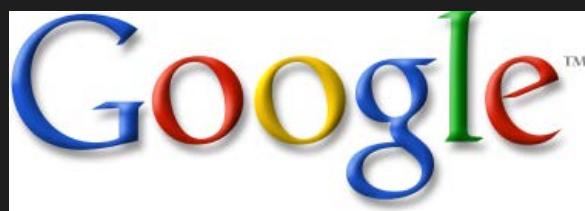
```
#git commit -a -m 'Add new target alias for doc index'
```

```
#git push origin HEAD:refs/for/master
```

Contemporary Code Reviews



facebook



Code Review Tools

Code Review Tools



Gerrit: <https://code.google.com/p/gerrit/>



Review Board: <https://www.reviewboard.org/>

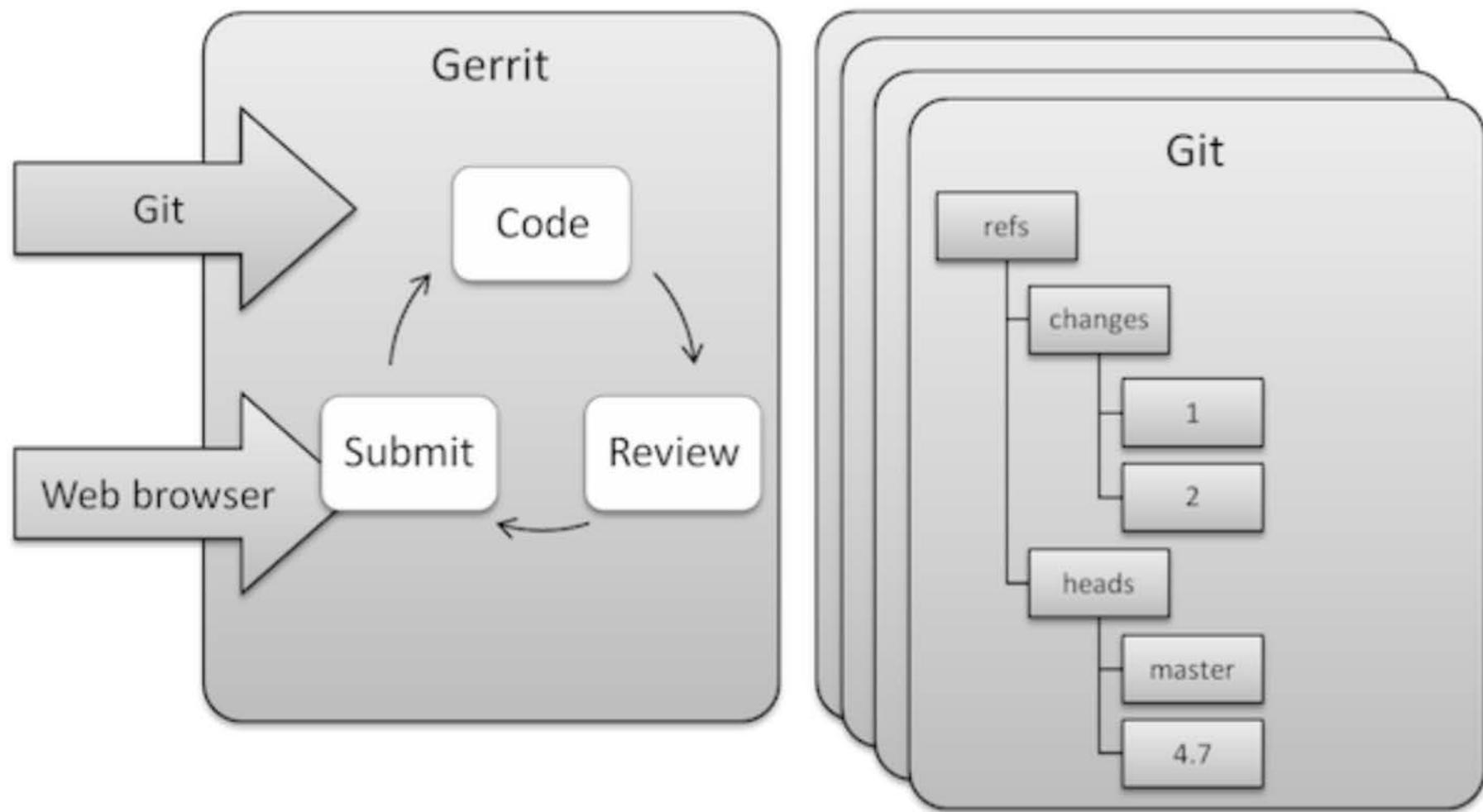


Phabricator: <https://phabricator.org/>

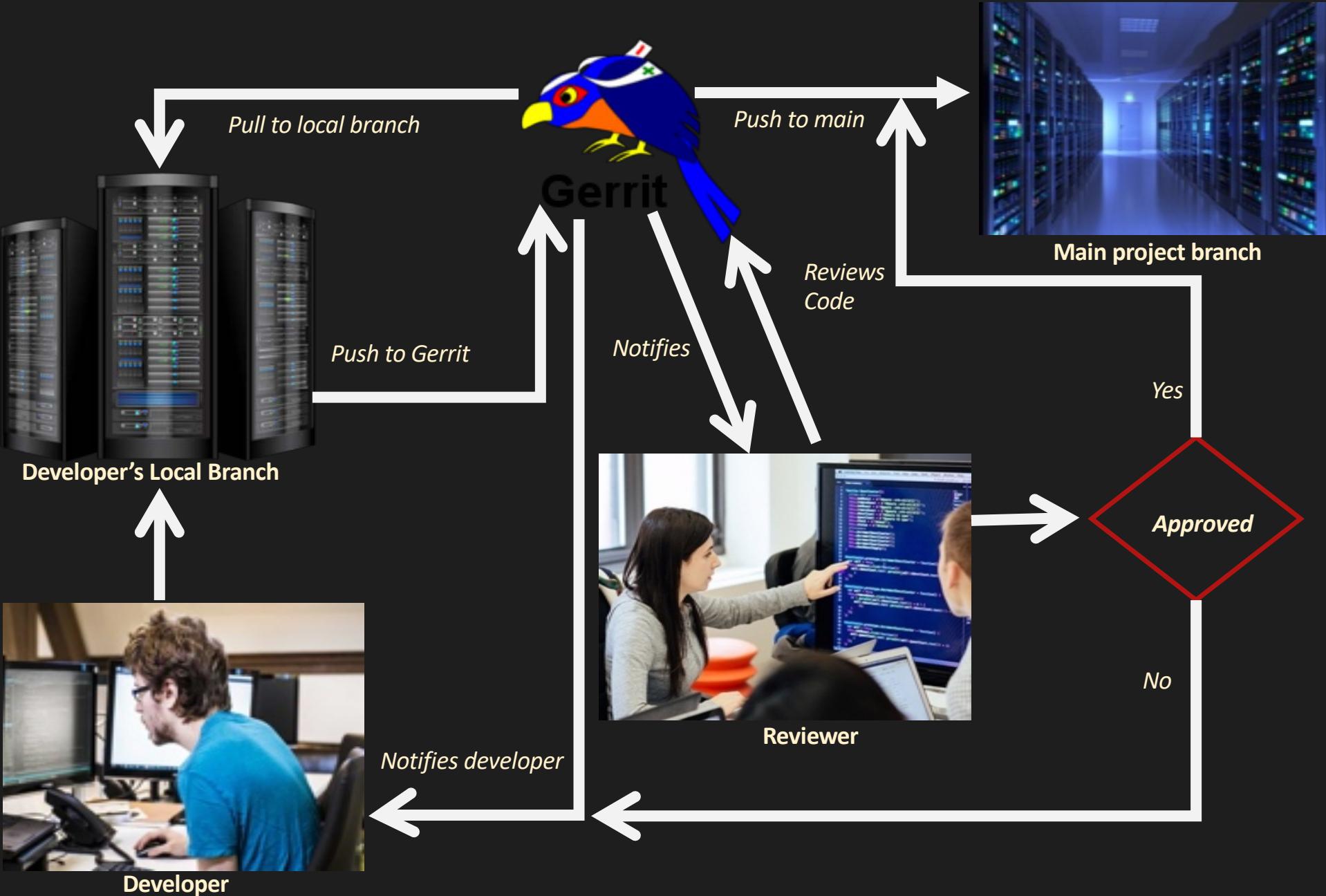


Crucible:
<https://www.atlassian.com/software/crucible>





Simplified Gerrit workflow

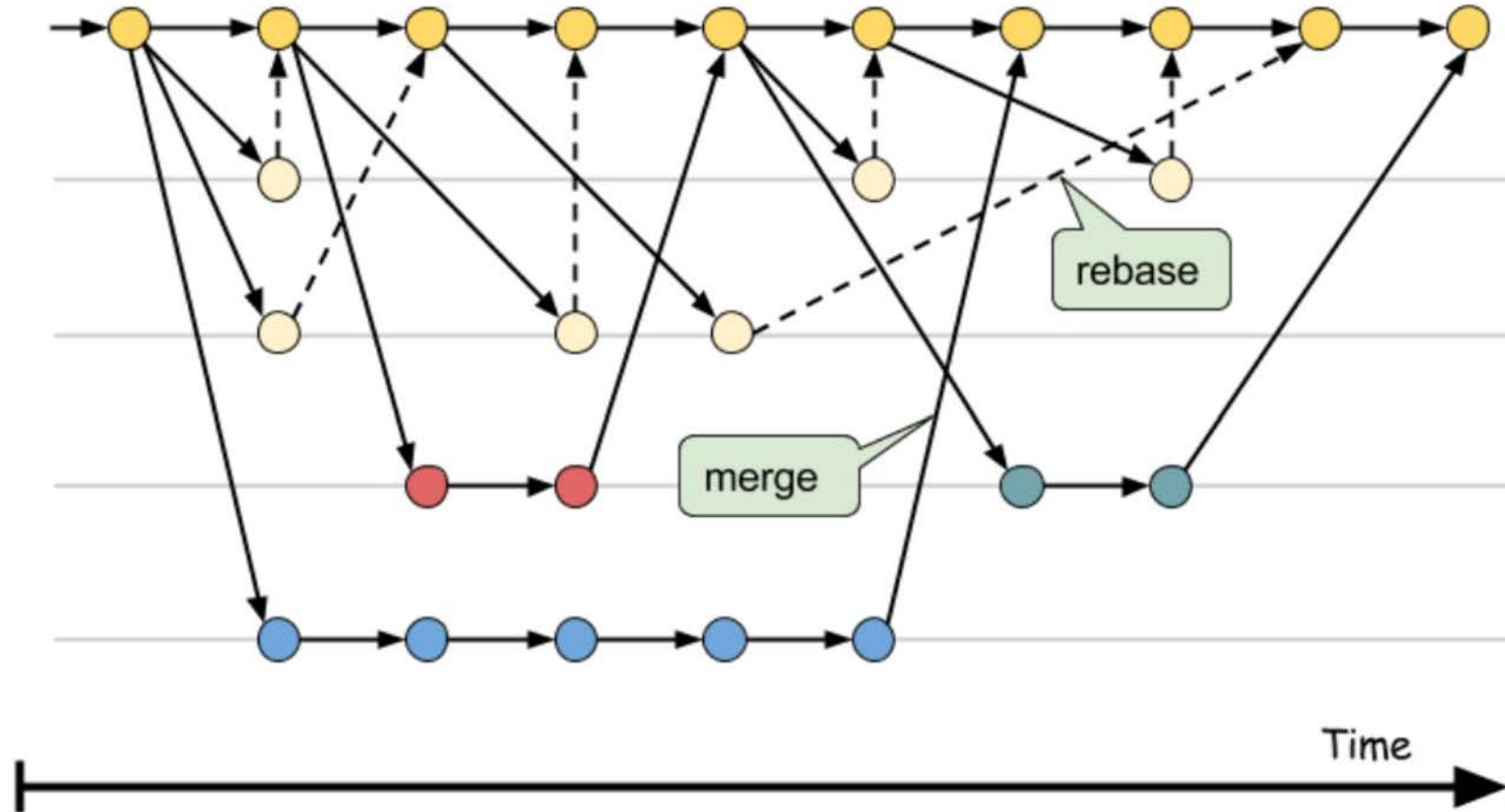


Gerrit Repository Structure

stable
branch

change
branches

feature
branches



Review Scores

Score	Description	Action
-2	This shall not be merged	Blocks submit
-1	I would prefer this is not merged as is	None
0	No score	None
+1	Looks good to me, but someone else must approve	None
+2	Looks good to me, approved	Allows submit

Gerrit Demo Video

<https://www.youtube.com/watch?v=jeWTvDad6VM>

Code Review Exercise

- Use your own code or download SampleCode.java from <https://SE4Science.org/tutorials/ECP19>
- Review the code

Feedback And Discussion

Feedback

<http://bit.ly/ECP-CR-Tutorial-Feedback>

References for further reading

- *Code Complete*, by Steve McConnell
- <https://www.codeproject.com/articles/524235/codeplusreviewplusguidelines>
- <https://blog.philippauer.de/code-review-guidelines>
- <https://github.com/joho/awesome-code-review>
- <https://www.planetgeek.ch/wp-content/uploads/2013/06/Clean-Code-V2.1.pdf>



Collaborate?

@SE4Science 

Jeffrey Carver
carver@cs.ua.edu



better
scientific
software
<http://BSSw.io>

Photo Credits

- <http://incolors.club/collectionfdwn-female-computer-programmer.htm>
- <http://tech.trivago.com/img/posts/code-review/code-review-3.jpg>
- <http://www.protectitip.com/wp-content/uploads/2014/11/Software-Code.jpg>
- http://www.computerhistory.org/atchm/wp-content/uploads/2013/11-marked_up_listing-542x404.jpg
- <https://static1.squarespace.com/static/53798babe4b0fca9449cf693/t/53f78774e4b0ce4d05e4152f/1408730997720/>
- <https://residentialwastesystems.com/wp-content/uploads/2016/10/dumpsters-trumbull-ct.jpg>
- <http://www.hipaasecurenow.com/index.php/beckers-hipaa-compliance-8-best-practices/>
- [https://commons.wikimedia.org/wiki/File:Collaboration_\(960_1759166\).jpg#metadata](https://commons.wikimedia.org/wiki/File:Collaboration_(960_1759166).jpg#metadata)
- <http://entertainment.time.com/2012/05/09/confessions-of-another-book-reviewer/>