Contemporary Peer Code Review Practices

Jeffrey C. Carver
Nasir U. Eisty

University of Alabama

Can you spot the mistakes?

```cpp
// file: IVR.CPP
void IVR()
{
        //press 1 for account balance, 2 for last transaction,
        //3 for last statement, any other for operator
        play_prompt();

        int key_pressed= get_user_choice();
        if(key_pressed ==1)
        {
                play_account_balance();
        }
        else if(key_pressed =2)
        {
                play_last_transaction();
        }
        else if(key_pressed ==3)
        {
                play_last_statement();
        }
        else transfer_to_operator();
}
```

Assignment operator instead of comparison

```java
//file: printer.java
if (user.isAuthenticated)
{
        userAccess = checkUserAuthorization(user);

        //if user has access to printer
        if(user.isAuthenticated && userAccess.printer)
                printUsageReport ();
        else
                emailUsageReport();
}
```

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                              80
#define   CDMA_SUBSCRIPTION_SOURCE_RUIM 1                               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);    83
static int set_cdma_subscription_source( AModem modem, ACdmaSubscriptionSource ss);             84
static int set_cdma_prl_version( AModem modem, int prlVersion);                                 85
                                                                                                86
```

```
            if (noMedia) {                                         478
                // In case the file is known and now is under a    479
                // .nomedia folder mark as not seen in order to    480
                // be removed from files table in the post scan.   481
            if (entry != null && noMedia) {                        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;                   483
            }                                                      484
            return null;                                           485
        }                                                          486
                                                                   487
```

```
    if ((nbytes=recvfrom(fd,msgbuf,MSGBUFSIZE,0,
62 »        »           »        (struct sockaddr *) &addr,&
63 »                     perror("recvfrom");
64 »                     exit(1);
65 »             }
66 »         puts(msgbuf);
```

[Name redacted]   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**

[Name redacted]   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]->p
   age_title ),
57                     wfMessage( 'articlefeedbackv5-special-goback' )->text());
```

Catrope   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']

TO ERR IS HUMAN

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.

And forgotten 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.

Typical Code Review Workflow

Requests Review

Reviews

Writes

Merge

Abandon

Mailing List Code Review

From	████████████████████████
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.

--

████████████
████████████████████████
████████████

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

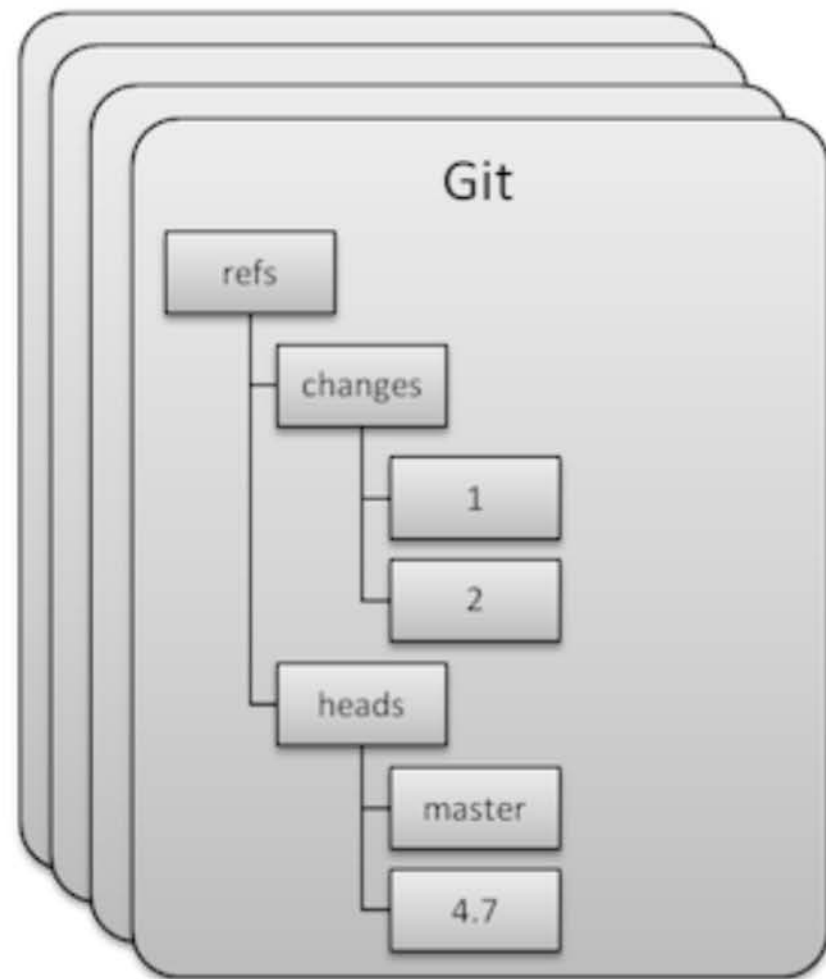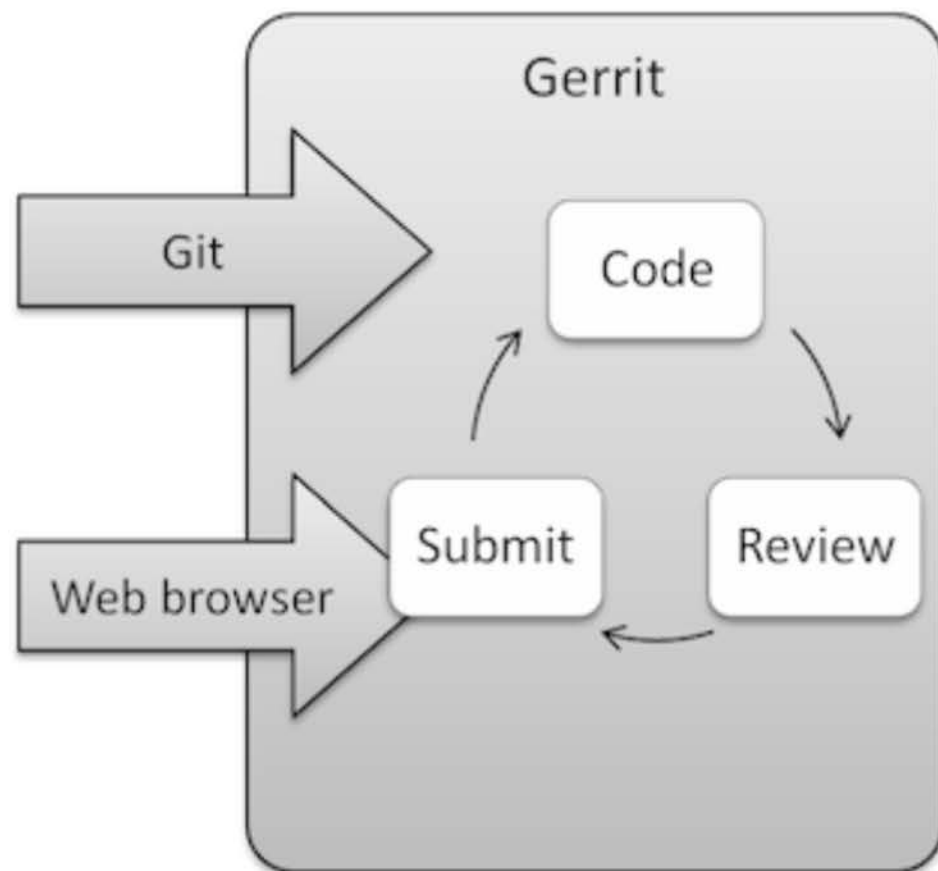# Contemporary Code Reviews
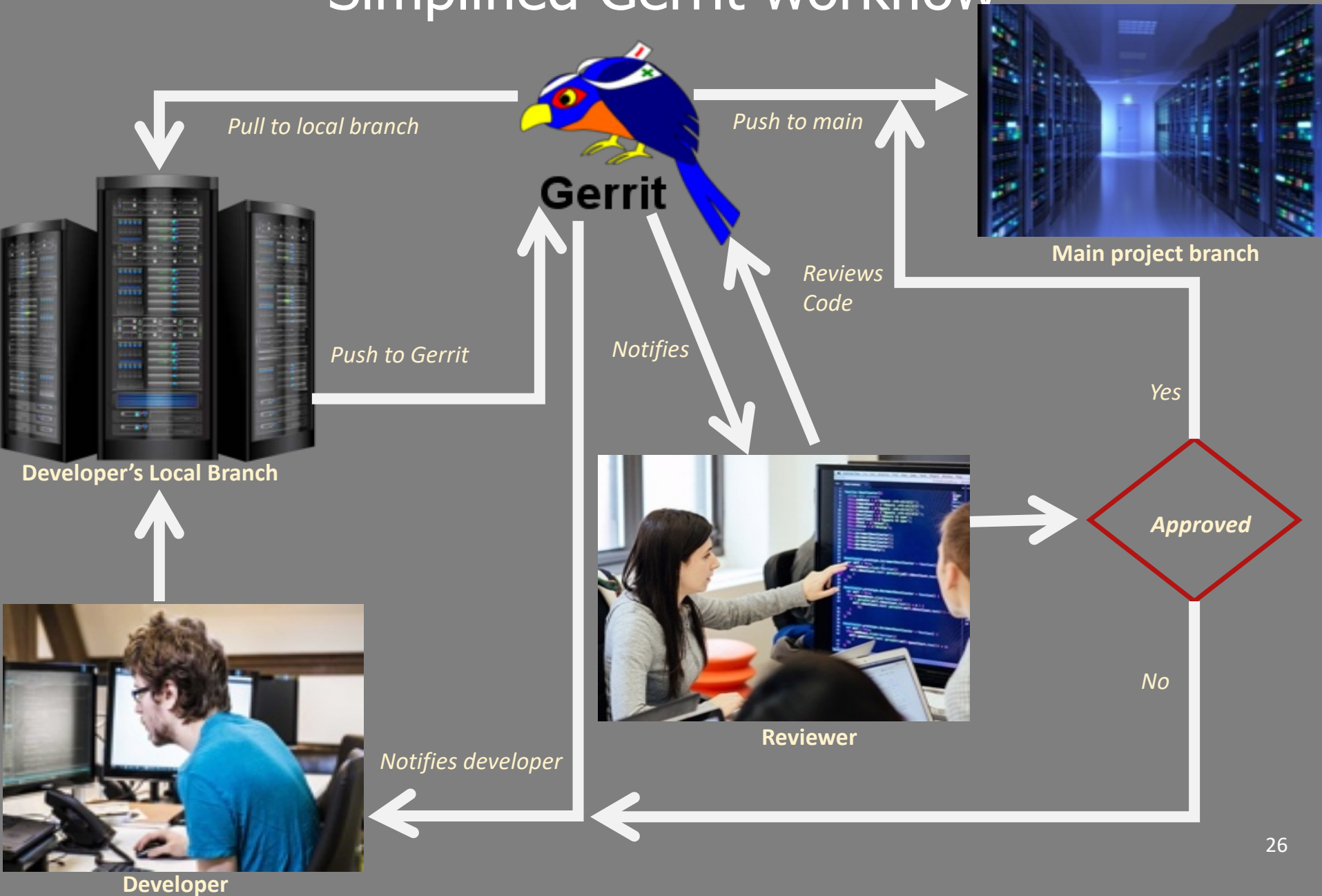
Issues Identified in code review

Benefits

Scientific Code Review

Gerrit

Simplified Gerrit workflow

Pull to local branch

Push to main

Main project branch

Push to Gerrit

Reviews Code

Notifies

Yes

Developer's Local Branch

Approved

No

Reviewer

Notifies developer

Developer
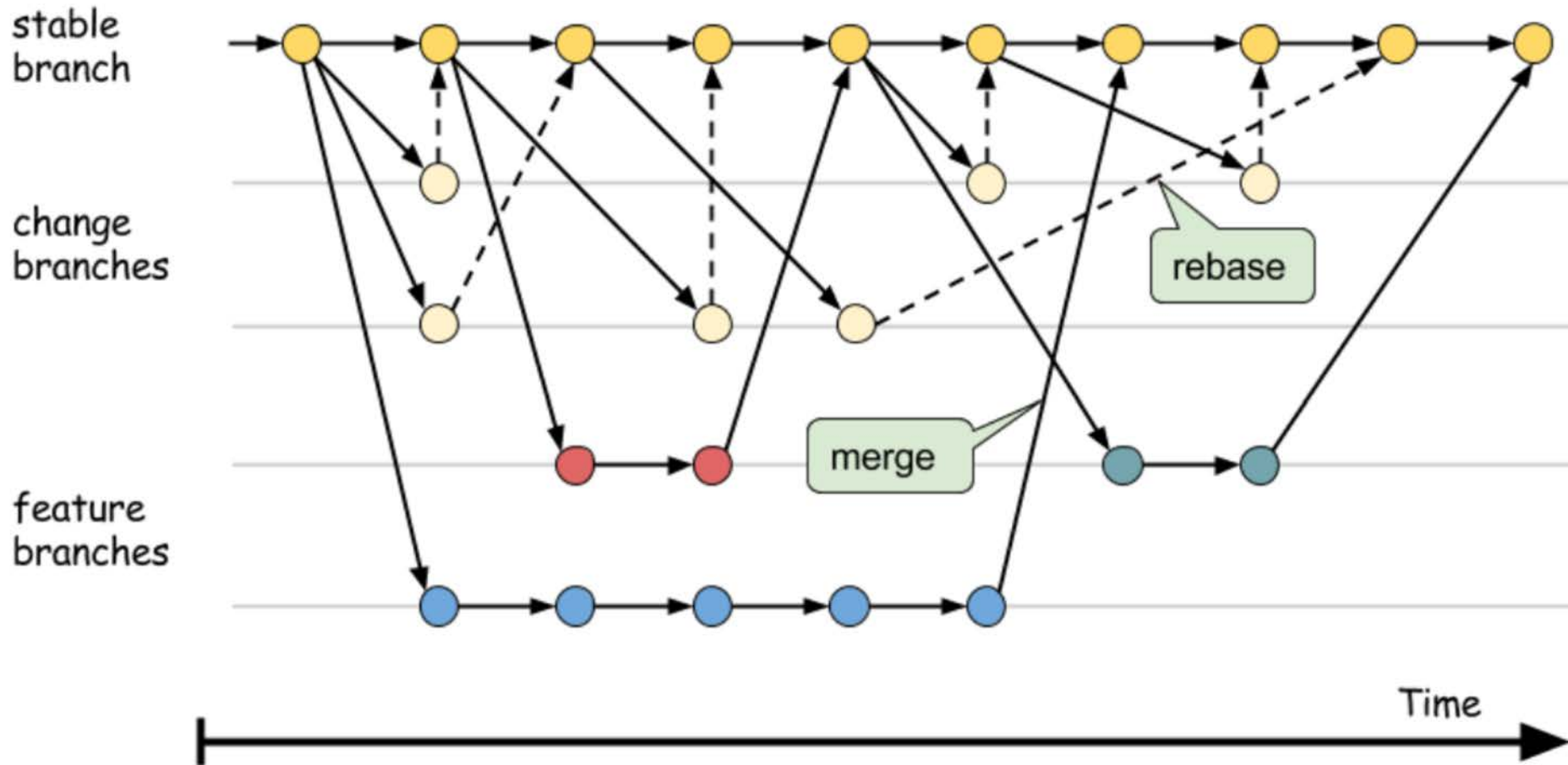
26

Gerrit Repository Structure

Gerrit Terminology

- Change:
  - The unit of review
  - Every git commit has a change-id
  - Results in a single commit when merged to the Git repository
  - Change numbers are unique and never change

- Patch Set:
  - A revision of a Change
  - Each time a Change is modified, it will receive a new patch set
  - Patch set numbering starts from 1
  - Technically, a patch set is a unique Git commit

Gerrit Terminology

- Label
  - A rating category that allows or blocks progress of a change. (i.e. Build, Deploy, Code Review)

- Score
  - The rating given to a label.
  - Code review spans +2 to -2

- Submit
  - Merges a change into the repository - until this happens, the code is outside of the repo

- Abandon
  - Archives a change (can be restored later)

- Project
  - A Git repository

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 Installation

http://bit.ly/Gerrit-Tutorial-SEA18

Code Review Exercise 1

- Pair with a neighbor

- Download dayOfYear.java from:
  http://SE4Science.org/CodeReview-SEA/

- Upload code to the "exercise"repository you created while setting up GerritHub

- Add your partner as a reviewer

- Review each other's code

Discussion

```java
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31 + 28 + 31 + 30;
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } else if (month == 7) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
    } else if (month == 8) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31;
    } else if (month == 9) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31;
    } else if (month == 10) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
    } else if (month == 11) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```

Don't repeat yourself

```
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
```

```
/** @param month   month of the year, where January=1 and December=12 [C1] */
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {        // we're in February   [C2]
        dayOfMonth += 31;    // add in the days of January that already passed [C3]
    } else if (month == 3) {
        dayOfMonth += 59;    // month is 3 here   [C4]
    } else if (month == 4) {
        dayOfMonth += 90;
    }

    ...
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth; // the answer   [C5]
}
```

```java
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } else if (month == 7) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
    } else if (            ) {
                                  + 30
    } else if (
        dayO                   + 31 + 30 + 31
    } else if (month == 10) {
        dayOfMonth += 31                    30 + 31 + 31 + 30;
    } else if (month ==
        dayOfMonth += 31                    + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```


Fail Fast

February 9, 2019

dayOfYear(1,9,2019)

dayOfYear(9,2,2019)

dayOfYear(2019,2,9)

37

```java
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31 + 28 + 31 + 30;
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } e
```

Fail Faster

dayOfYear(String month, int dayOfMonth, int year) {

 …

 }

```java
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
    } else if (month == 11) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```

```java
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31 + 28 + 31 + 30;
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } else if (month == 7) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
    } else if (month == 8) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31;
    } else if (month == 9) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31;
    } else if (month == 10) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
    } else if (month == 11) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```

Avoid Magic Numbers

```java
public static int dayOfYear(int month, int dayOfMonth, int year) {
    if (month == 2) {
        dayOfMonth += 31;
    } else if (month == 3) {
        dayOfMonth += 59;
    } else if (month == 4) {
        dayOfMonth += 90;
    } else if (month == 5) {
        dayOfMonth += 31 + 28 + 31 + 30;
    } else if (month == 6) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31;
    } else if (month == 7) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30;
    } else if (month == 8) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31;
    } else if (month == 9) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31;
    } else if (month == 10) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30;
    } else if (month == 11) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31;
    } else if (month == 12) {
        dayOfMonth += 31 + 28 + 31 + 30 + 31 + 30 + 31 + 31 + 30 + 31 + 31;
    }
    return dayOfMonth;
}
```
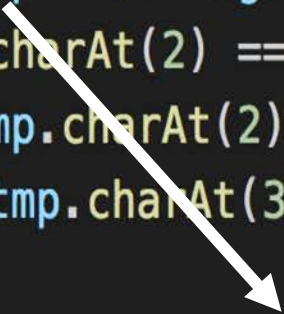
One Purpose for each variable

Code Review Exercise 2

- Keep the same pair

- Download leapYear.java from: http://SE4Science.org/CodeReview-SEA/

- Upload the code to the "exercise"repository you created while setting up GerritHub

- Add your partner as a reviewer

- Review each other's code

Use Good Names

```java
public static boolean leap(int y) {
    String tmp = String.valueOf(y);
    if (tmp.charAt(2) == '1' || tmp.charAt(2) == '3' || tmp.charAt(2) == 5
        || tmp.charAt(2) == '7' || tmp.charAt(2) == '9') {
        if (tmp.charAt(3)=='2'||tmp.charAt(3)=='6') return true; /*R1*/
        else
```

secondsPerDay = 86400

```java
                                             At(3) == '0') {
            return false; /*R3*/
        }
        if (tmp.charAt(3)=='0'||tmp.charAt(3)=='4'||tmp.charAt(3)=='8')
            return true; /*R4*/
    }
    return false; /*R5*/
}
```

```java
public static boolean leap(int y) {
    String tmp = String.valueOf(y);
    if (tmp.charAt(2) == '1' || tmp.charAt(2) == '3' || tmp.charAt(2) == 5
        || tmp.charAt(2) == '7' || tmp.charAt(2) == '9') {
        if (tmp.charAt(3)=='2'||tmp.charAt(3)=='6') return true; /*R1*/
        else
            return false; /*R2*/
    }else{
        if (tmp.charAt(2) == '0' && tmp.charAt(3) == '0') {
            return false; /*R3*/
        }
        if (tmp.charAt(3)=='0'||tmp.charAt(3)=='4'||tmp.charAt(3)=='8')
            return true; /*R4*/
    }
    return false; /*R5*/
}
```

Use Whitespace to Help the Reader

```java
public static bo
    String tmp = String.valueOf(y);
    if (tmp.charAt(2) == '1' || tmp.charAt(2) == '3' || tmp.charAt(2) == 5
            || tmp.charAt(2) == '7' || tmp.charAt(2) == '9') {
        if (tmp.charAt(3)=='2'||tmp.charAt(3)=='6') return true; /*R1*/
        else
            return false; /*R2*/
    }else{
        if (tmp.charAt(2) == '0' && tmp.charAt(3) == '0') {
            return false; /*R3*/
        }

        if (tmp.charAt(3)=='0'||tmp.charAt(3)=='4'||tmp.charAt(3)=='8')
            return true; /*R4*/
    }
    return false; /*R5*/
}
```

45

```java
public static boolean leap(int y)
    String tmp = String.valueOf(y);
    if (tmp.charAt(2) == '1' || tmp.charAt(2) == '3' || tmp.charAt(2) == 5
        || tmp.charAt(2) == '7' || tmp.charAt(2) == '9') {
        if (tmp.charAt(3)=='2'||tmp.charAt(3)=='6') return true; /*R1*/
        else
            return false; /*R2*/
    }else{
        if (tmp.charAt(2) == '0' && tmp.charAt(3) == '0') {
            return false; /*R3*/
        }

        if (tmp.charAt(3)=='0'||tmp.charAt(3)=='4'||tmp.charAt(3)=='8')
            return true; /*R4*/
    }
    return false; /*R5*/
}
```

Other 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

Feedback And Discussion

Feedback

http://bit.ly/SEA-CR-Tutorial-Feedback

Code Review

BEST PRACTICE

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_(9601759166).jpg#metadata