

Contemporary Peer Code Review Practices

Jeffrey C. Carver
Nasir U. Eisty

University of Alabama

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

Jaime A Lopez-Sollano

Sep 23, 2010

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

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

~~David Stinson~~

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

~~David Stinson~~

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]->page_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'](#)

A hand is holding a white rectangular sign with a thin black border. The sign contains the text "TO ERR IS HUMAN" in a bold, black, sans-serif font. The hand is visible on the left side of the sign, with the thumb and index finger gripping the edge. The background is plain white.

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



Writes

Reviews

Merge



Abandon



PAGE	111	SHEP	APPLE	DOS		PAGE
3573						WBOOT
3574	3755	ADF137	LDA	IBBUFP+1	; GET START OF DOS	
3575	3758	BDE337	STA	BONDOS	; SAVE IT	
3576	3758	3B	SEC			
3577	375C	ADE737	LDA	ADOSLD+1	; CALCULATE	
3578	375F	EDE337	SBC	BONDOS		
3579	3762	BDE037	STA	NDPDS	; NO DOS PAGES	
3580	3765	BDE237	STA	IBDSEC		<i>more</i>
3581						
3582	3768	A900	LDA	#0	; TRACK=0	
3583	376A	BDE037	STA	IBTRK	; TRACK=0	
3584	376D	BDE037	STA	IBSECT	; SECTOR=0	
3585	3770	BDF037	STA	IBBUFP		
3586						
3587	3773	ADE737	LDA	ADOSLD+1	; GET BOOT START ADR	
3588	3776	BDF137	STA	IBBUFP+1	; TO BUFP	
3589	3779	BDFE36	STA	GRSPQ	; TO GARBAGE RECORD	<i>0</i>
3590						
3591	377C	A90A	LDA	#10	; NO OF BOOT PAGES	
3592	377E	BDE137	STA	BRWCNT	; TO BOOT I/O COUNTER	
3593	3781	0A	ABLA		; AND	
3594	3782	0A	ASLA	-LDR #48	; TO	
3595	3783	0A	ASLA	NOCD	; GARBAGE RECORD	
3596	3784	BDFE36	STA	GRPOC		
3597						
3598	3787	A902	LDA	#IBCHTS	; SET WRITE	
3599	3789	BDF437	STA	IBCHD		
3600						
3601	378C	209F37	JSR	BOOTIO	; GO WRITE BOOT SECTORS	
3602						
3603	378F	ADE337	LDA	BONDOS	; SET START OF DOS	
3604	3792	BDF137	STA	IBBUFP+1		
3605						
3606	3795	ADE037	LDA	NDPDS		
3607	3798	BDE137	STA	BRWCNT		
3608	379B	209F37	JSR	BOOTIO	; GO WRITE DOS	
3609						
3610	379E	60	RTS		; DONE	
3611						

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]
Senior System Administrator
[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
```

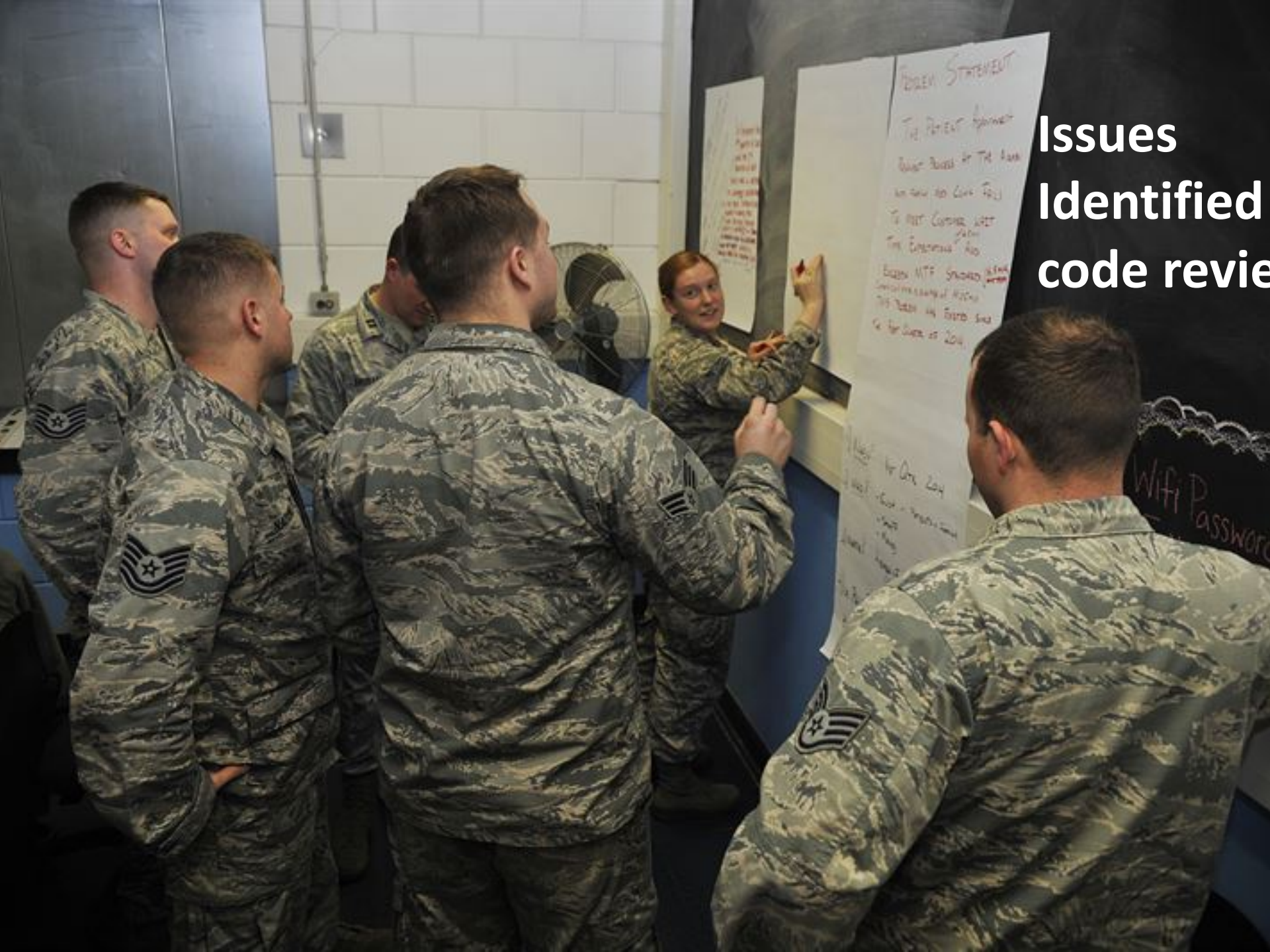
```
#git commit -a -m 'Add new target alias for doc index'
```

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

Contemporary Code Reviews



Issues Identified code review



BODEN STATEMENT

The Project Approval
Basic Rules of The Code
An idea for Code 2011
To Not Continue with
The "Business" Also
Boden MTF Standard
The Rules of Code 2011
Code for Code of 2011

Wifi Password

for Code 2011
Code - 2011-2011
Code - 2011-2011
Code - 2011-2011



Benefits

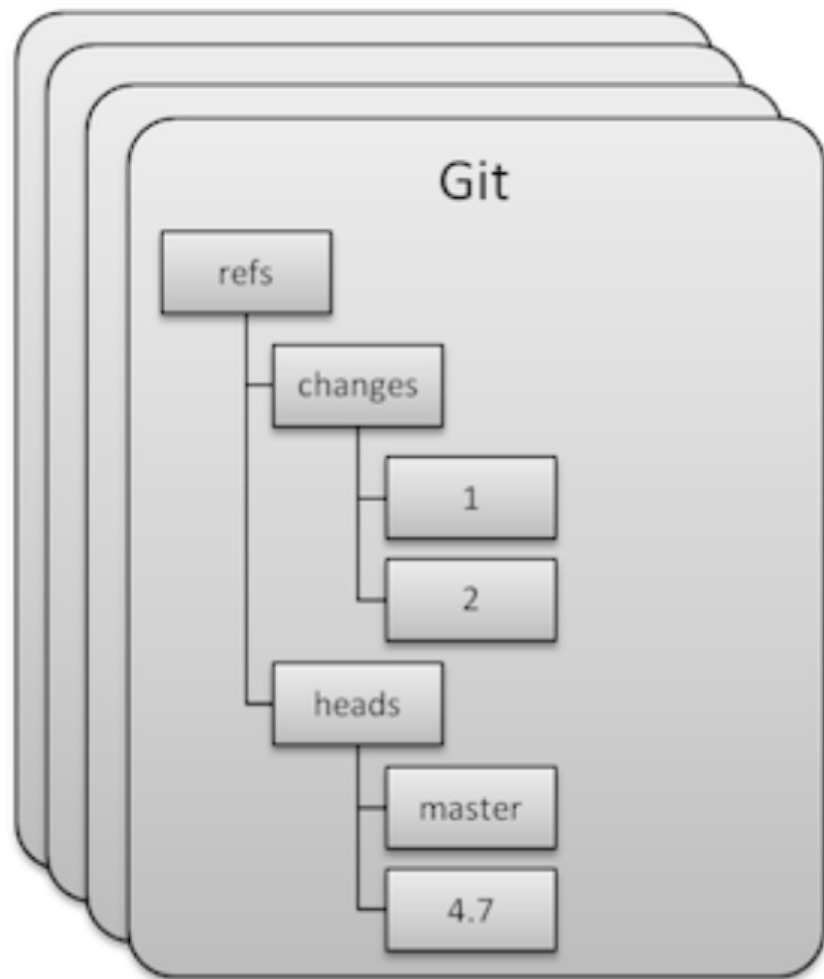
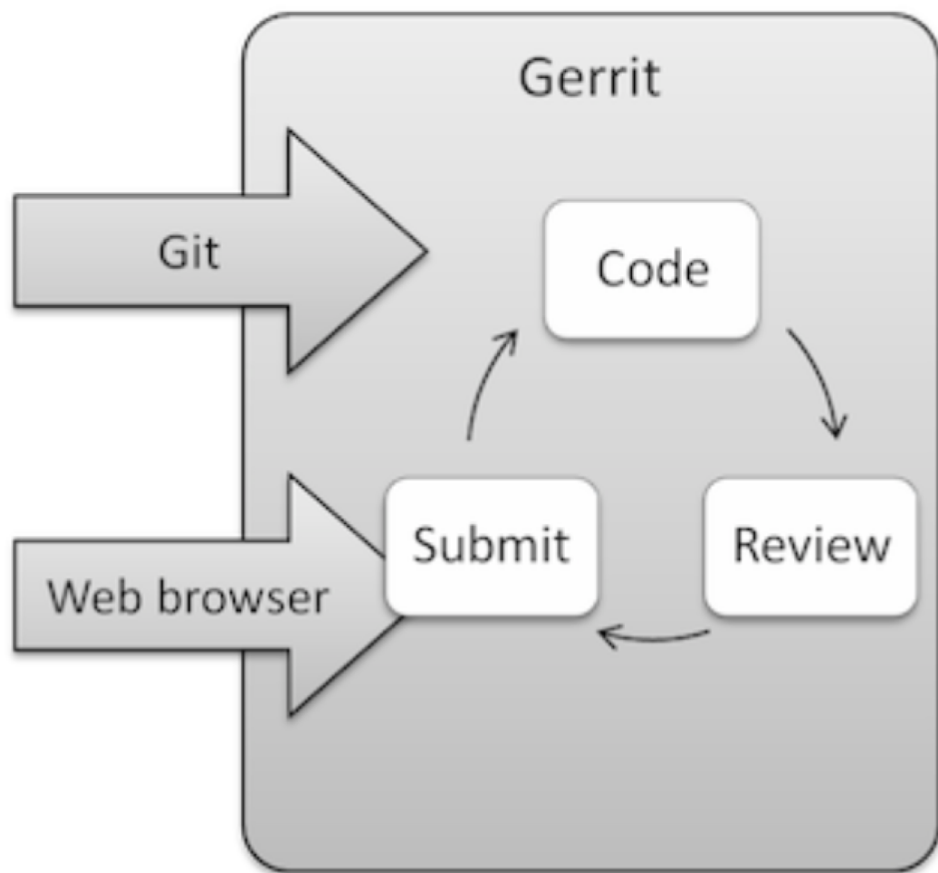
Scientific Code Review

WMS

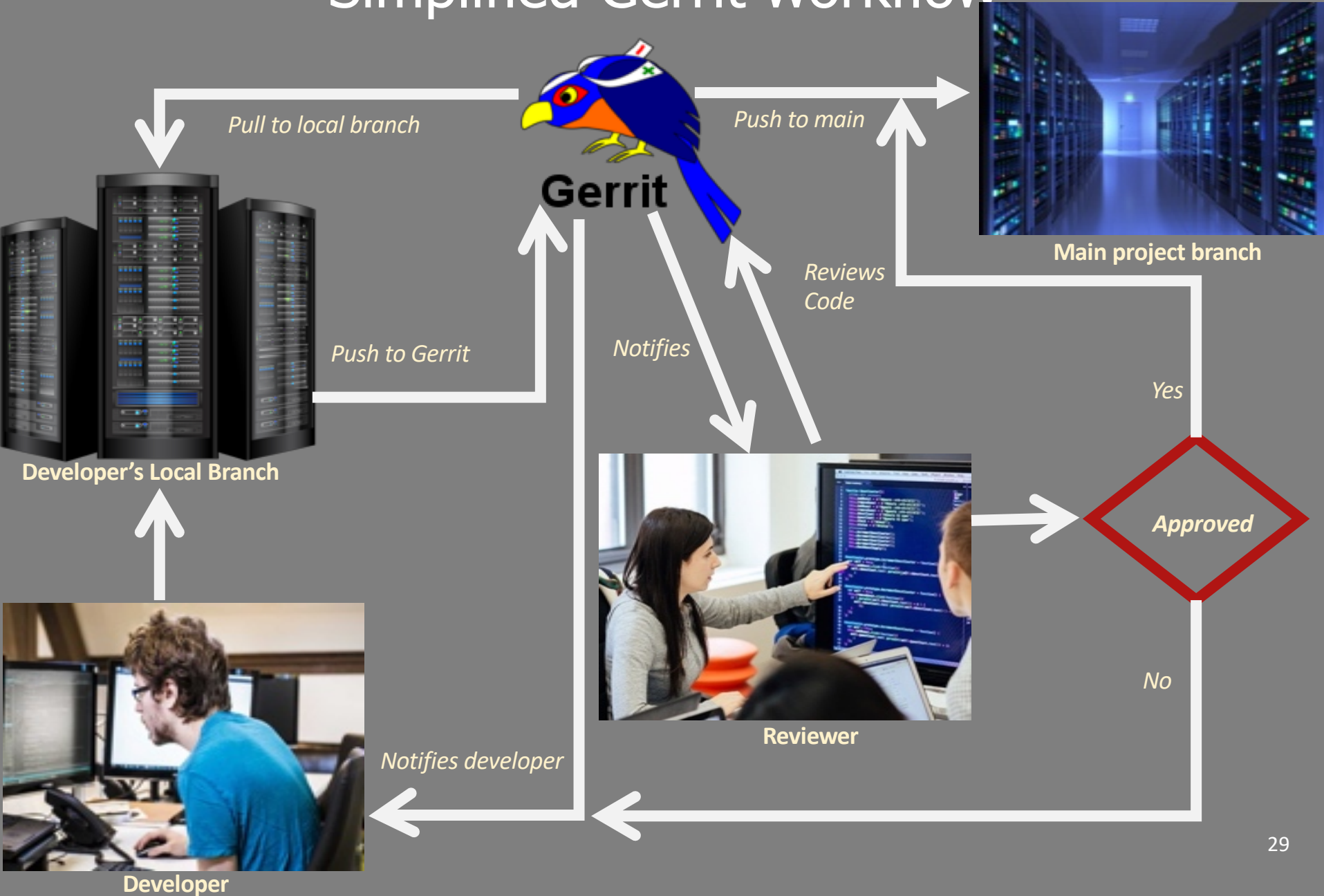




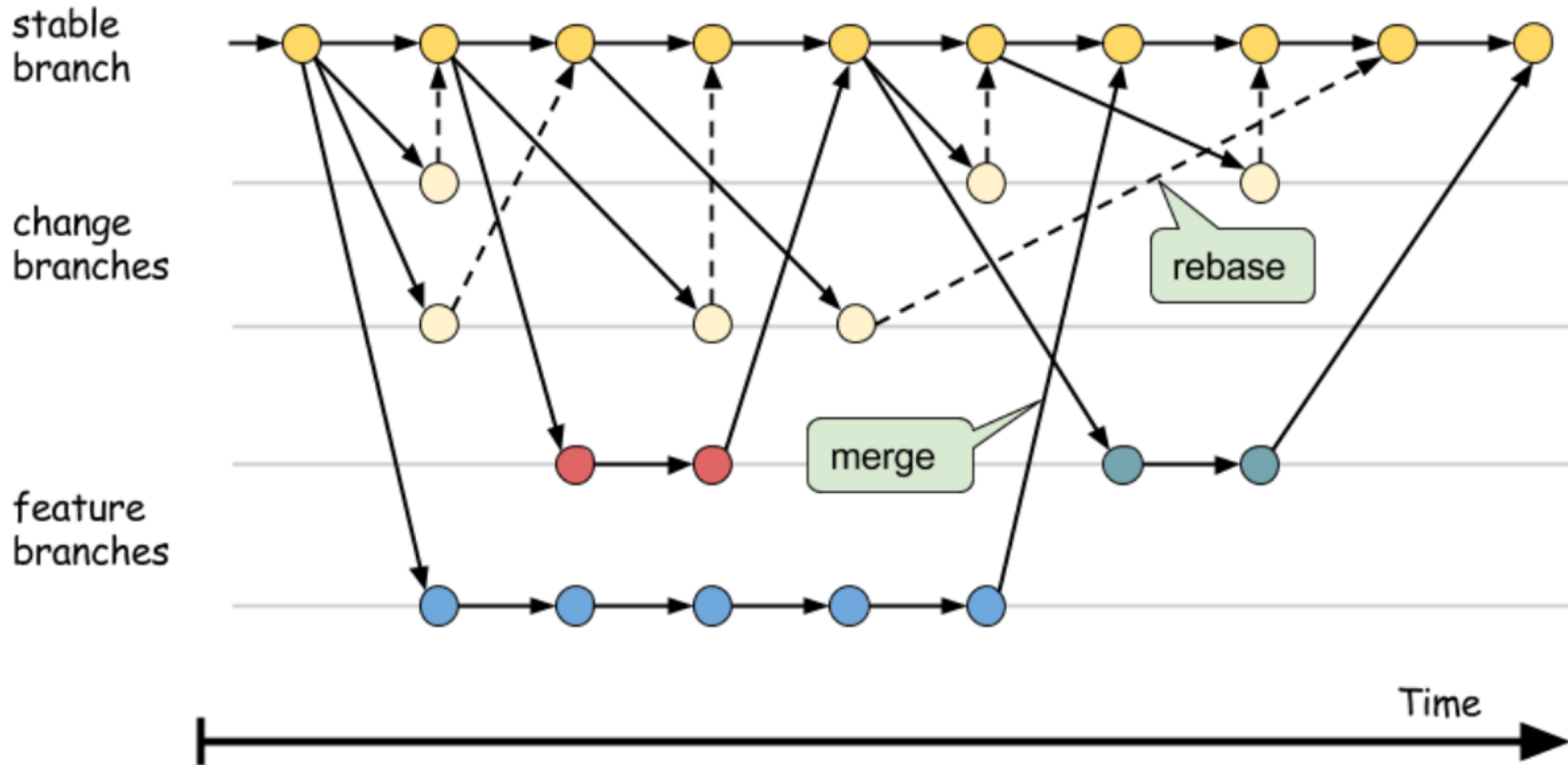
Gerrit



Simplified Gerrit workflow



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

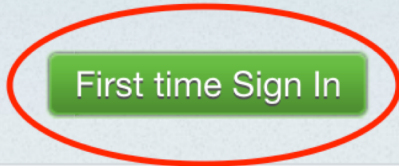
Review Scores

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-eScience18>


- Sign in to <https://github.com>
- Create two new repositories-
 - 'exercise1' & 'exercise2'
- Go to <http://gerrithub.io>




Gerrit Code Review with GitHub repositories

GitHub is the most popular free Git hosting in the world with over 5 Millions repositories! You can now use it with the power of **Gerrit**, the best **Enterprise Engine** for Git and Code Review.

 **Sign in with GitHub**
Use your existing GitHub account for logging in with Gerrit.

 **Import repositories**
Get your code hosted on GitHub visible and replicated in Gerrit.

 **Code review workflow**
Implement the Code Review using Gerrit workflow User-Experience.

 **GitHub pull requests**
Keep in touch with external users synchronizing pull requests with reviews.

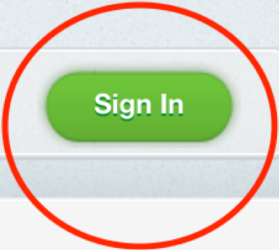


1 **Sign-in to GitHub**
Login to Gerrit using your GitHub credentials: **you need to authorize GerritHub to access your account**, keys and repositories. With a **single click** you can start using **Gerrit Code Review**.

2 **Replicate Git repositories**
Gerrit automatically replicates all the changes back and forth to GitHub. Allows to **integrate with Community-based OpenSource** project whilst keeping a **single Enterprise Git repository in Gerrit**.

3 **Pull requests with Gerrit**
Fetch your GitHub pull requests and upload them in Gerrit for review. Enable a **single point of validation** of the Code Review workflow and keep your **external contributors up-to-date** on GitHub.

Sign In using GitHub credentials >>



What should I do for getting started ?

- 1. **Sign In** with your GitHub username and password
- 2. **Allow GerritHub** accessing your profile
- 3. Select the **SSH Public Keys** you want to import
- 4. Select the **Git repositories** you want to use

Why you need to login using GitHub ?

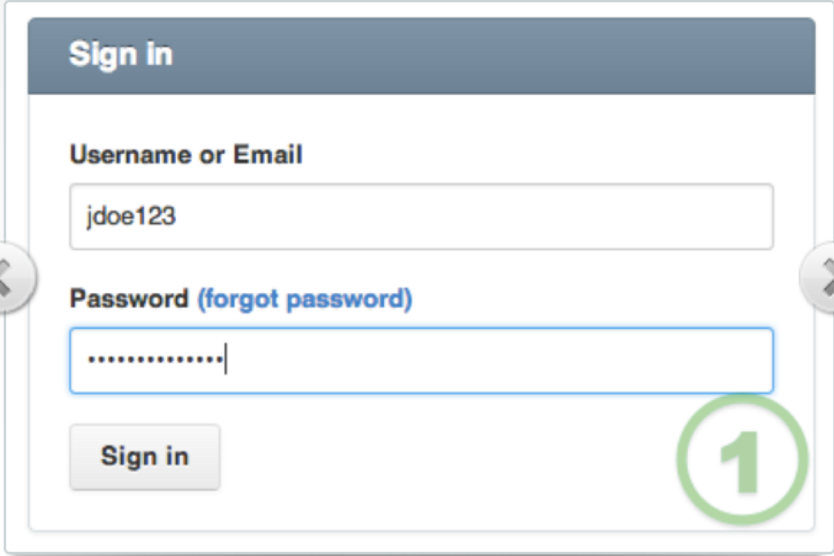
GerritHub and GitHub work together to give you Code Review on your existing repositories: your identity on Gerrit is synchronized with your GitHub identity.

Why GerritHub requires write permissions ?

GerritHub needs a **user:email+public_repo scopes** when authenticating with the GitHub API: the only information read from your profile are **your full name, e-mail and SSH public keys**: we do NOT write any information to your GitHub profile or followers or share with any other service on-line.

When start using GerritHub, your **Git commits and patches for review will be mirrored back to GitHub** thanks to Gerrit replication mechanism. **GerritHub pushes on your behalf your reviewed commits to your GitHub repositories.**

In order to get the replication work properly, you should **push to Gerrit** and use your **GitHub repository for pull and pull-requests only**. If you keep on **pushing directly to GitHub** you would cause the Gerrit **replication process to stop**: this is done for avoiding to overwrite the GitHub history because of the potential conflicts.



Login Scope Selection

Cancel

Login >

Which level of GitHub access do you need?

DEFAULT

Review and contribute to public repos using your e-mail identity and organizations' membership

Allow to: Access e-mail addresses , Read and push to public repos , Read membership to public/private organizations

Reviewer

Review other people's changes using only your e-mail identity (EU Citizens for GDPR)

Allow to: Access e-mail addresses

Private

Review and contribute to private and public repos using your e-mail identity and organizations' membership

Allow to: Read/write profile info , Read and push to public and private repos

Remember my choice



Authorize GerritHub



GerritHub by [gerritforge-ltd](#)

wants to access your **tutorialtest43** account



Personal user data

Email addresses (read-only)



Repositories

Public only



Organizations and teams

Read-only access



Authorize gerritforge-ltd

Authorizing will redirect to
<https://review.gerrithub.io>

Account profile

Cancel

Next >

GitHub account details

Username:

Full name:

e-Mail:

Import GitHub SSH Public Keys



[Account](#) > Repositories Replication

Cancel


Import >

Select GitHub repositories to clone and replicate

tutorialtest57 / Repository name

Filter by name

10 Rows

 Not seeing your organizations or repositories? [Login with a different GitHub Scope](#) and try again.

PUBLIC tutorialtest57 / [exercise1](#)

PUBLIC tutorialtest57 / [exercise2](#)

[Account](#) > Repositories Replication

Cancel

Next >

Select GitHub repositories to clone and replicate



tutorialtest57



/ Repository name

Filter by name

10



Rows



Not seeing your organizations or repositories? [Login with a different GitHub Scope](#) and try again.

PUBLIC

[tutorialtest57 / exercise1](#)



Done: repository replicated to Gerrit.

PUBLIC

[tutorialtest57 / exercise2](#)



Done: repository replicated to Gerrit.



Cancel

Next >

tutorialtest57

10

Rows

First

Previous

Next

Last

Search pull request

Filter repositories



Id

Repository / Pull
request subject /
Author

Timestamp

Commits

0

All repositories

No data available in table



Subject	Status	Owner	Assignee	Project	Branch
Work in progress					
No changes					
Outgoing reviews					
No changes					
Incoming reviews					
No changes					
Recently closed					
No changes					

tutorialtest57
ttestemail57@gmail.com

- [Settings](#)
- [Sign out](#)

- [Profile](#)
- [Preferences](#)
- [Diff Preferences](#)
- [Menu](#)
- [Change Table Colu...](#)
- [Notifications](#)
- [Email Addresses](#)
- [HTTP Credentials](#)
- [SSH Keys](#)
- [Groups](#)
- [Mail Filters](#)

User Settings

Profile

ID	1015254
Email	ttestemail57@gmail.com
Registered	12:41 PM
Username	tutorialtest57
Full name	<input type="text" value="tutorialtest57"/>
Status	<input type="text"/>

Preferences

Changes per page	<input type="text" value="25 rows per page"/>
Date/time format	<input type="text" value="Jun 3 ; Jun 3, 2016"/> <input type="text" value="4:10 PM"/>
Email notifications	<input type="text" value="Only comments left by others"/>
Email format	<input type="text" value="HTML and plaintext"/>
Diff view	<input type="text" value="Side by side"/>
Expand inline diffs	<input type="checkbox"/>
Publish comments on push	<input type="checkbox"/>
Set new changes to "work in progress" by default	<input type="checkbox"/>

- [Profile](#)
- [Preferences](#)
- [Diff Preferences](#)
- [Menu](#)
- [Change Table Colu...](#)
- [Notifications](#)
- [Email Addresses](#)
- [HTTP Credentials](#)
- [SSH Keys](#)
- [Groups](#)
- [Mail Filters](#)

HTTP Credentials

Username tutorialtest57

[GENERATE NEW PASSWORD](#)

SSH keys

Comment

Status Public key

[SAVE CHANGES](#)

New SSH key

New SSH Key

[ADD NEW SSH KEY](#)

Groups

Name	Description	Visible to all
Anonymous Users		No
kubedge-admin	people who can manage kubedge projects	No
Registered Users		No
TestGroup	asdfasdf	No
user/tutorialtest57		No
userid/tutorialtest57		No

Mail Filters

Gerrit emails include metadata about the change to support writing mail filters.

Here are some example Gmail queries that can be used for filters or for searching through archived messages. View the [Gerrit documentation](#) for the complete set of footers.

Name Query

- [Profile](#)
- [Preferences](#)
- [Diff Preferences](#)
- [Menu](#)
- [Change Table Colu...](#)
- [Notifications](#)
- [Email Addresses](#)
- [HTTP Credentials](#)
- [SSH Keys](#)
- [Groups](#)
- [Mail Filters](#)

HTTP Credentials

Username tutorialtest57

[GENERATE NEW PASSWORD](#)

SSH keys

Comment

Status Public key

[SAVE CHANGES](#)

New SSH key

New SSH Key

[ADD NEW SSH KEY](#)

New Password:

E/f7z0fM9QDCtH0FtEg6p+zeNZ7JlIn1vYeNkYaNvng

*This password will not be displayed again.
If you lose it, you will need to generate a new one.*

[CLOSE](#)

TestGroup	asdfasdf	No
user/tutorialtest57		No
userid/tutorialtest57		No

Mail Filters

Gerrit emails include metadata about the change to support writing mail filters.

Here are some example Gmail queries that can be used for filters or for searching through archived messages. View the [Gerrit documentation](#) for the complete set of footers.

Name Query



Subject	Status	Owner	Assignee	Project	Branch	Updated
Work in progress						
No changes						
Outgoing reviews						
No changes						
Incoming reviews						
No changes						
Recently closed						
No changes						



Projects

Filter:

Groups

Project Name	Project Description
007-surajit/Directory-List-PhoneGap-Plugin	A plugin to list contents of application directory for iOS and Android Cordova apps
01org/ciao	Ciao - Cloud Integrated Advanced Orchestrator
091387-ea975-lab3/ea975-lab3	Lab 3 - Gerrit + Github
0intro/plan9-contrib	Plan 9 from Bell Labs, with contributions.
0s4l/cloaked-archer	
0s4l/cloaked-archer-2	
0s4l/dsf	
0x707/g_train2	
0x707/gerrit_train	
0x707/t_train	training
100591487/Lab-1	Lab 1 files
118DBK/GerritTest	GerritTest
11NJ/frameworks_support	
11NJ/git-repo	
11NJ/ldd3-examples-3.x	port ldd3 source code examples to linux 3.x
11NJ/spring-oauth-server	
11NJ/springapp	
11NJ/springrest	



Projects

Filter:

Groups

Project Name	Project Description	Repository Browser	Re on
tutorialtest57/exercise1		(GitHub)	
tutorialtest57/exercise2		(GitHub)	



- [Projects](#)
- [tutorialtest57/exerci...](#)
- [Access](#)
- [Commands](#)
- [Branches](#)
- [Tags](#)
- [Groups](#)

tutorialtest57/exercise1

Download

[ANONYMOUS HTTP](#) **HTTP** [SSH](#)

Clone with commit-msg hook

```
git clone https://tutorialtest57@review.gerrithub.io/a/tutorialtest57/exercise1 && (cd exercise1
```

[COPY](#)

Clone

```
git clone https://tutorialtest57@review.gerrithub.io/a/tutorialtest57/exercise1
```

[COPY](#)

Configurations

Description

<Insert project description here>

Project Options

- State:
- Submit type:
- Allow content merges:
- Create a new change for every commit not in the target branch:
- Require Change-Id in commit message:
- Reject implicit merges when changes are pushed for review:
- Enable adding unregistered...

Gerrit Review UI

Code Review Exercise 1

- Download dayOfYear.java from:
<https://se4science.org/tutorials/escience18>
- Upload code to the “exercise1” repository you created while setting up GerritHub
- Review the code

Discussion

```
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) {

```

Comments Where
Needed

```

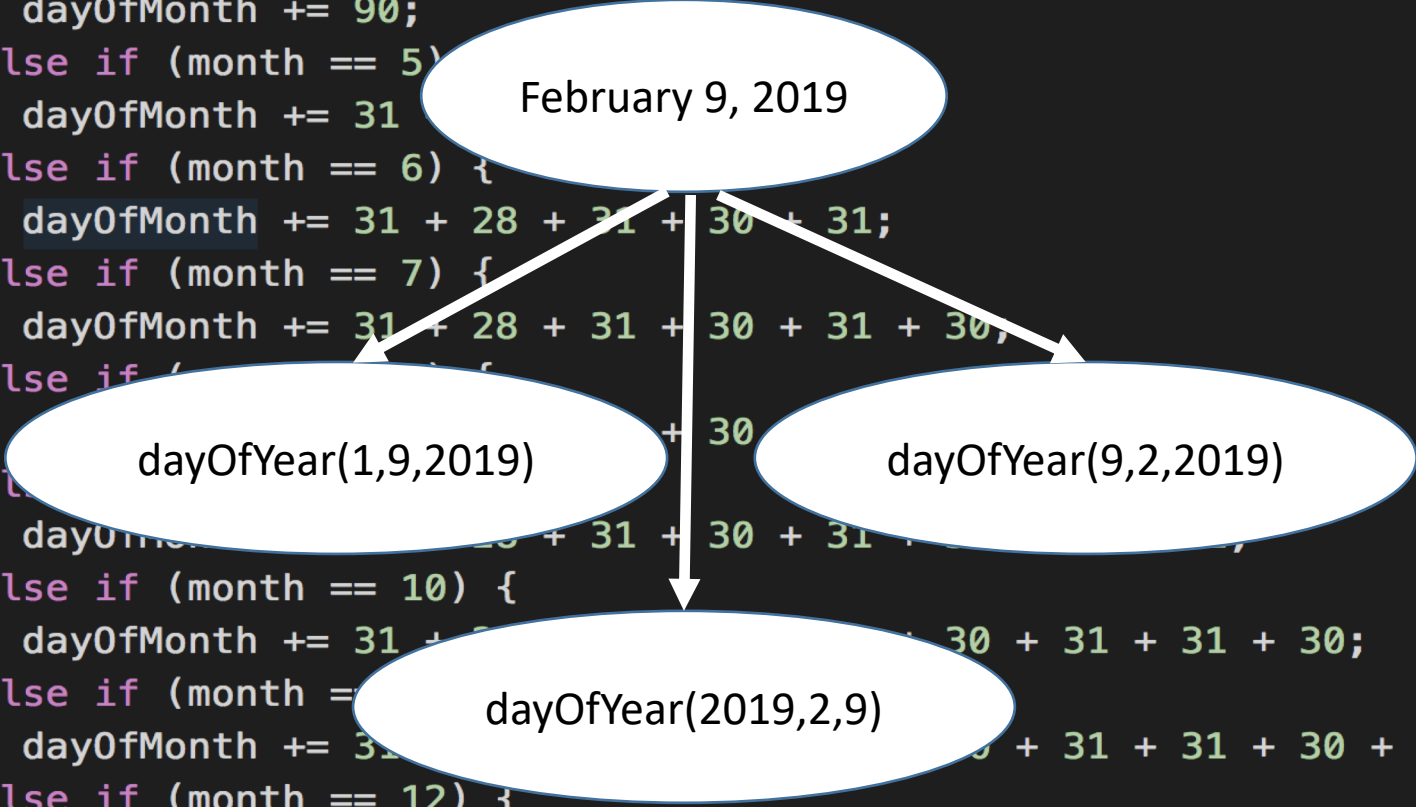
    /** @param month  month of the year, where January=1 and Dec
    ember=12 [C1] */
    public static int dayOfYear(int month, int dayOfMonth, int year) {
        if (month == 2) {           // we're in February [C2]
            dayOfMonth += 31;      // add in the days of January that
            already 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]
    }
}

```

+ 31;

```
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 += 121;
    } else if (month == 6) {
        dayOfMonth += 151;
    } else if (month == 7) {
        dayOfMonth += 181;
    } else if (month == 8) {
        dayOfMonth += 212;
    } else if (month == 9) {
        dayOfMonth += 243;
    } else if (month == 10) {
        dayOfMonth += 273;
    } else if (month == 11) {
        dayOfMonth += 304;
    } else if (month == 12) {
        dayOfMonth += 334;
    }
    return dayOfMonth;
}
```

Fail Fast



```

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
} e
} e
} e
    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;
}

```

Fail Faster

```

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

```

```

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

```

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

- Download leapYear.java from:
<https://se4science.org/tutorials/escience18>
- Upload the code to the
“exercise2” repository you created while
setting up GerritHub
- Review the code

Use Good Names

```
public static boolean leap(int y) {  
    String tmp = String.valueOf(y);
```

- methodsAreNamedWithCamelCaseLikeThis
- variablesAreAlsoCamelCase
- CONSTANTS_ARE_IN_ALL_CAPS_WITH_UNDERSCORES
- ClassesAreCapitalized
- packages.are.lowercase.and.separated.by.dots

Magic Numbers

```
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

```
public static boolean  
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*/  
}
```

Bugs hidden in this code

```
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/eScience-CR-Tutorial-Feedback>

Code Review

BEST
PRACTiCE



Collaborate?



@SE4Science 

Jeffrey Carver
carver@cs.ua.edu

<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/atcm/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.hipaasecurenw.com/index.php/beckers-hipaa-compliance-8-best-practices/>
- [https://commons.wikimedia.org/wiki/File:Collaboration_\(9601759166\).jpg#metadata](https://commons.wikimedia.org/wiki/File:Collaboration_(9601759166).jpg#metadata)