

# Contributing to WordPress Core

Jonathan Desrosiers  
@Desrosj



# About Me

- Been using WordPress for 8+ years.
- Plugin developer, Theme developer, everything in between.
- Core contributor (credited in 4.4, 4.2, 4.1, 3.7, and 3.6)

# About Me

- Web Developer III at Boston University
- [bu.edu/id](https://bu.edu/id)



# Congratulations!

You are taking initiative!

# Congratulations!

WordPress would not be what it is today without  
contributions from people like you.

# What took you so long?

- Too busy.
- Don't know how.
- I am not really an expert.
- I'm intimidated.

# Zip It



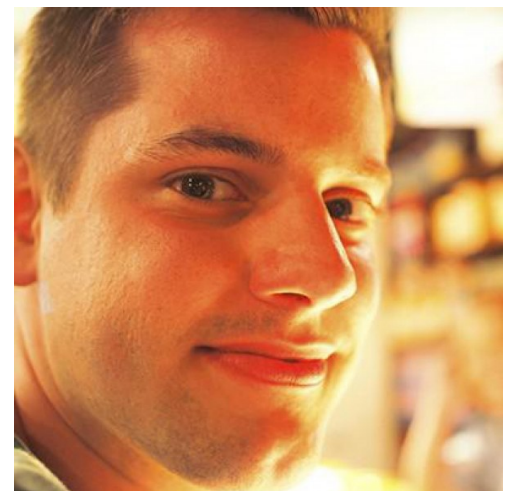
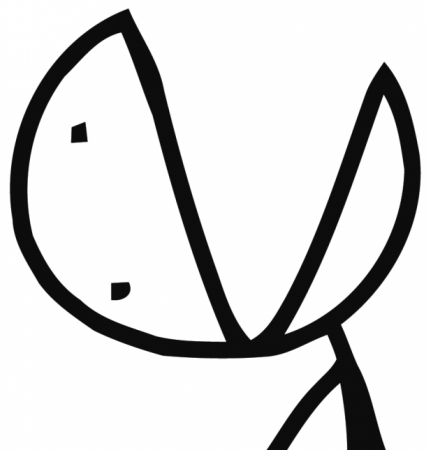
# What took you so long?

Every. Little. Thing. Counts.



# Everyone Starts Somewhere!

How did current leaders in the WordPress community start contributing to core?



# John Blackbourn (Johnnbillion)

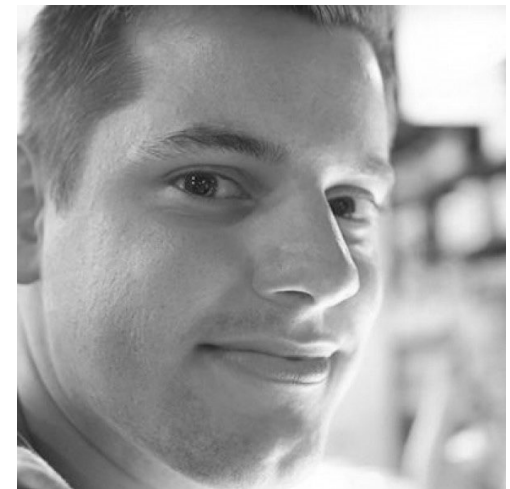
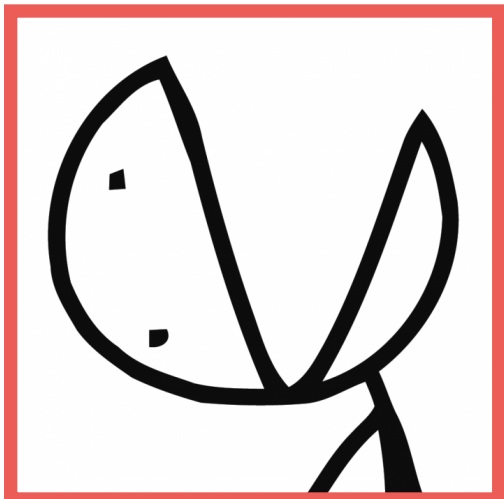
## 4.1 Release Lead

☐ Unmodified ☒ Added ☐ Removed

trunk/wp-admin/link-add.php

Tabular | Unified

r4664	r4999	
29	29	<div id="wp-link-bookmarklet" class="wrap">
30	30	<h3><?php _e('Add Link Bookmarklet'); ?></h3>
31		<p><?php _e('Right click on the following link and choose "Bookmark This Link..." to create an add link shortcut. Right now this only works on Mozilla or Netscape, but we&#8217;re working on it.');
	31	<p><?php _e('Right click on the following link and choose &#0147;Bookmark This Link...&#0148; or &#0147;Add to Favorites...&#0148; to create a Link This shortcut.');
32	32	<?php printf('<p><a href="%s" title="'.__( 'Link add bookmarklet' ).'>'.__( 'Link This' ).'</a></p>', "javascript:void(linkmanpopup=window.open('" . get_option('siteurl') . "/wp-admin/link-add.php? action=popup&amp;linkurl='+escape(location.href)+'&amp;name='+escape(document.title), 'LinkManager', 'scrollbars=yes,width=750,height=5 ?>
33	33	</div>





# Drew Jaynes (drewapicture)

## 4.2 Release Lead

☐ Unmodified ☒ Added ☐ Removed

[trunk/wp-admin/includes/media.php](#)

Tabular | Unified

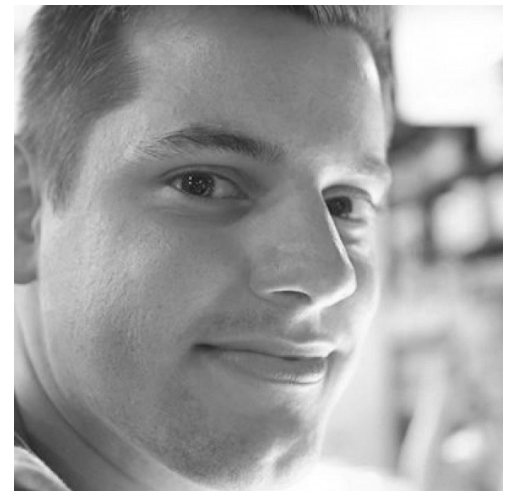
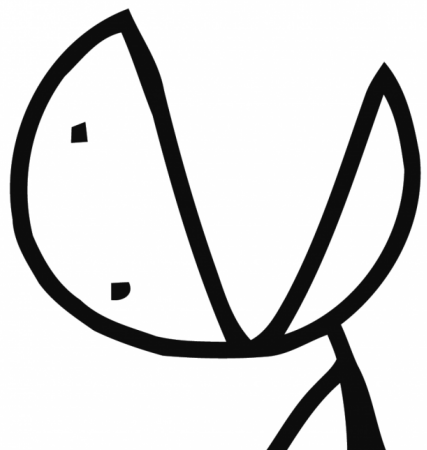
r19268	r19275
1498	1498
1499	1499
1500	
1501	
	1500
	1501
	1502
	1503
1502	1504
1503	1505

```
html = '';

if ( f.url.value )
    html = '<a href="' + f.url.value + '">' + html + '</a>';

if ( f.url.value ) {
    url = f.url.value.replace(/'/g, '&#039;').replace(/"/g, '&quot;').replace(/</g, '&lt;').replace(/>/g, '&gt;');
    html = '<a href="' + url + '">' + html + '</a>';
}

if ( caption )
```



# Konstantin Obenland (Obenland)

## 4.3 Release Lead

☐ Unmodified ☒ Added ☐ Removed

trunk/wp-admin/custom-background.php		Tabular	Unified
r20358	r20757		
384	384	if ( isset( \$_REQUEST['context'] ) && \$_REQUEST['context'] == 'custom-background' ) {	
385	385	\$form_fields = array( 'image-size' => \$form_fields['image-size'] );	
386		\$form_fields['buttons'] = array( 'tr' => '<tr class="submit"><td></td><td><a data-attachment-id="' . \$post->ID .	
		'" class="wp-set-background">' . __( 'Set as background' ) . '</a></td></tr>' );	
	386	\$form_fields['buttons'] = array( 'tr' => '<tr class="submit"><td></td><td><a data-attachment-id="' . \$post->ID .	
		'" class="wp-set-background">' . __( 'Set as background' ) . '</a></td></tr>' );	
387	387	\$form_fields['context'] = array( 'input' => 'hidden', 'value' => 'custom-background' );	
388	388	}	

trunk/wp-admin/custom-header.php		Tabular	Unified
r20719	r20757		
956	956	), admin_url( 'themes.php' ) );	
957	957		
958		\$form_fields['buttons'] = array( 'tr' => '<tr class="submit"><td></td><td><a data-location="' . \$href . ' "	
		class="wp-set-header">' . __( 'Set as header' ) . '</a></td></tr>' );	
	958	\$form_fields['buttons'] = array( 'tr' => '<tr class="submit"><td></td><td><a data-location="' . \$href . ' "	
		class="wp-set-header">' . __( 'Set as header' ) . '</a></td></tr>' );	
959	959	\$form_fields['context'] = array( 'input' => 'hidden', 'value' => 'custom-header' );	
960	960	}	





# Scott Taylor (wonderboymusic)

## 4.4 Release Lead

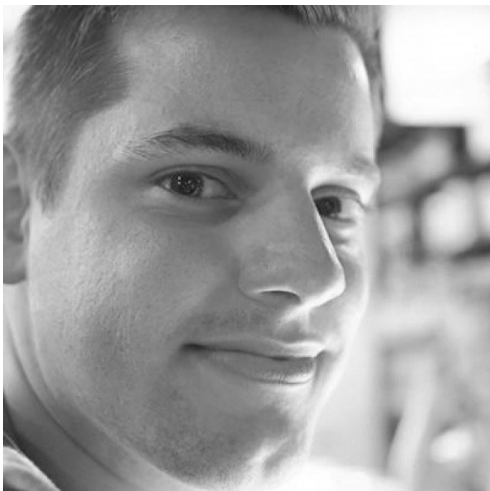
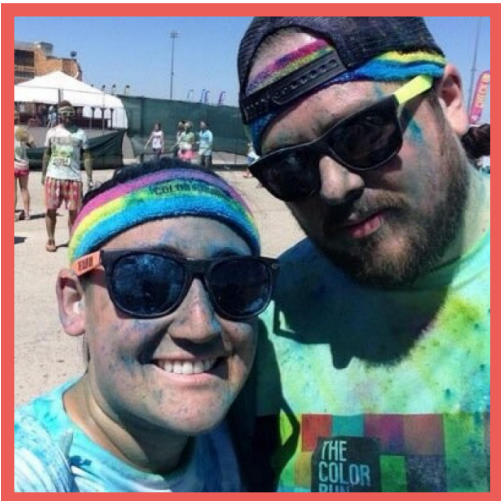
☐ Unmodified   ☒ Added   ☐ Removed

[trunk/wp-settings.php](#)

Tabular | Unified

r18532	r18549
30	30
31	31
32	32
33	33
	32
	33
34	34
35	35

```
// Disable magic quotes at runtime. Magic quotes are added using wpdb later in wp-settings.php.  
set_magic_quotes_runtime( 0 );  
@ini_set( 'magic_quotes_sybase', 0 );  
@ini_set( 'magic_quotes_runtime', 0 );  
@ini_set( 'magic_quotes_sybase', 0 );  
  
// Set default timezone in PHP 5.
```



# Dominik Schilling (ocean90)

## 4.6 Release Lead

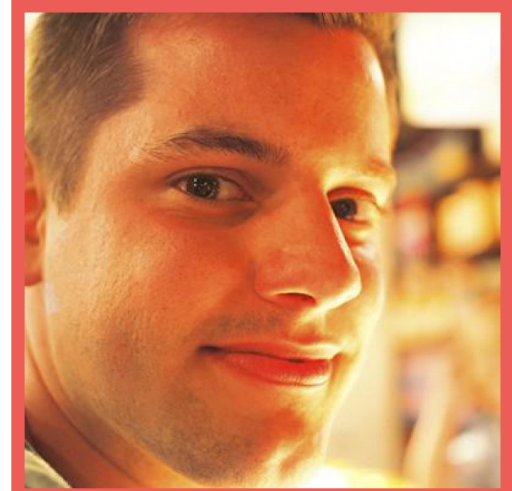
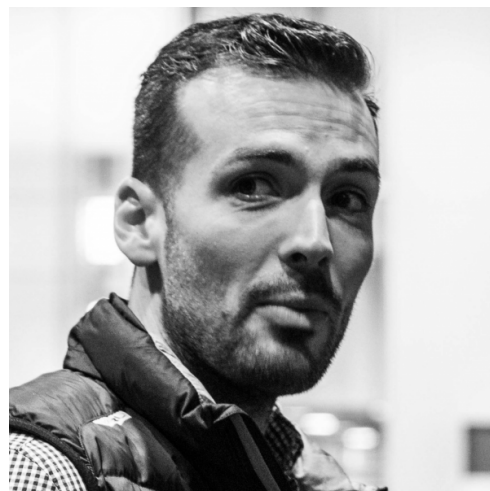
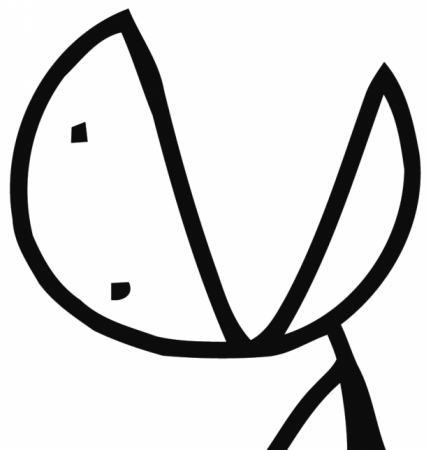
☐ Unmodified ☒ Added ☐ Removed

trunk/wp-admin/includes/dashboard.php

Tabular | Unified

r13256	r13373
650	650
651	651
652	
	652
653	653
654	654

```
echo '<p><label for="comments-number">' . __('Number of comments to show:') . '</label>';  
echo '<input id="comments-number" name="widget-recent-comments[items]" type="text" value="' . $number . '" size="3" />  
<small>' . __(' (at most 15)') . '</p>';  
echo '<input id="comments-number" name="widget-recent-comments[items]" type="text" value="' . $number . '" size="3" />  
<small>' . __(' (at most 15)') . '</small></p>';  
}
```



# Jonathan Desrosiers (Desrosj)

## Future Release Lead?

☐ Unmodified ☒ Added ☐ Removed

trunk/wp-admin/css/wp-admin.css

Tabular | Unified

r24578	r24588
--------	--------

3979	3979
------	------

}

3980	3980
------	------

	3981
--	------

```
.form-wrap .form-field #parent {
```

```
    max-width: 100%;
```

	3982
--	------

```
}
```

	3983
--	------

	3984
--	------

3981	3985
------	------

```
.col-wrap h3 {
```

3982	3986
------	------

```
    margin: 12px 0;
```



# Everyone Starts Somewhere

Little changes add up to make WordPress great!



# What we'll cover Today

- Discuss the many ways to contribute to WordPress.
- Talk about how the WordPress Core development process works.
- Discover the tools you will need to become familiar with.
- Dive in and get our hands dirty with a little code.
- Make WordPress contributions!


# Make WordPress - Contributor Groups

- Make WordPress ([make.wordpress.org](https://make.wordpress.org)) is a set of blogs that relate to the different contributor groups that help develop WordPress.

## Make WordPress

Whether you're a budding developer, a pixel-perfect designer, or just like helping out, we're always looking for people to help make WordPress even better.

If you want to get involved in WordPress, this is the place to be. We've got blogs for each contributor group, general news, and upcoming events.



### There are many different ways for you to get involved with WordPress:

#### <> Core

The core team makes WordPress. Whether you're a seasoned PHP developer or are just learning to code, we'd love to have you on board. You can write code, fix bugs, debate decisions, and help with development. [Learn more about Core »](#)

*Weekly chats: Wednesdays @ 20:00 UTC  
#core on Slack*

#### 🎨 Design

The design group is focused on the designing and developing the user interface. It's a home for designers and UXers alike. There are regular discussions about mockups, design, and user testing. [Learn more about Design »](#)

*Weekly chats: Thursdays @ 17:00 UTC  
#design on Slack*

#### 📱 Mobile

The mobile team builds the iOS, Android, Windows, and

#### ★ Accessibility

The accessibility group provides accessibility expertise across the

# Community Team

If you're interested in organizing a meetup or a WordCamp, the community blog is a great place to get started. There are groups working to support events, to create outreach and training programs, and generally support the community.

# Support Team

The Support Team is responsible for all things support related.

- Answer support questions in the forums on [WordPress.org](https://wordpress.org/support/)
- Answer questions in [WordPress.org](https://wordpress.org/slack/) Slack

Everyone knows the answer to something!

# Documentation Team

The Documentation Team is responsible for all things documentation, including the Codex, handbooks, [developer.wordpress.org](https://developer.wordpress.org), admin help, inline docs, and other general wordsmithing across the WordPress project.

- Update and edit the Codex
- Helping out with the handbooks
- Updating WordPress' inline documentation
- Helping to build [developer.wordpress.org](https://developer.wordpress.org)

# Training Team

The Training Team creates downloadable lesson plans and related materials for instructors to use in a live workshop environment.

- Writing – Create lesson plans.
- Copyediting – Check lesson plans for grammar, spelling and punctuation. Make sure they align with our style guide.
- Testing – Run tests of our beta lessons at Meetups or workshops and give us feedback.
- Auditing – Check content for accuracy with the current version of WordPress

# TV Team

The TV team reviews and approves every video submitted to WordPress.tv. They also help WordCamps with video post-production and are responsible for the captioning and subtitling of published videos.

- Post-production
- Subtitles/captions

# Polyglot Team

WordPress is used all over the world and in many different languages. If you're a polyglot, help out by translating WordPress into your own language.

- Suggest text translations at [translate.wordpress.org](https://translate.wordpress.org)



# Theme Review Team

The Theme Review Team reviews and approves every Theme submitted to the WordPress Theme repository. Reviewing Themes sharpens your own Theme development skills.

# Accessibility Team

The a11y team provides accessibility expertise across the project. They make sure that WordPress core and all of WordPress' resources are accessible.

# Themes & Plugins

Create your own WordPress Themes & Plugins, and submit/release them in the [WordPress.org](https://wordpress.org) repositories.

# Make WordPress - Contributor Groups

See? There are a ton of ways to get involved, even if you have little or no knowledge of code!

Let's learn about how WordPress core is developed and maintained.



# WordPress Release Cycle

- WordPress version releases (X.X) run on 4 month release cycles (3 releases per year).
- Minor bug fix and security releases (X.X.X) happen whenever necessary in between version releases.
- Deadlines are not arbitrary, even if they force things to be trimmed.

# WordPress Release Cycle - Philosophies

- We're never done with simplicity. We want to make WordPress easier to use with every single release.
- The core of WordPress will always provide a solid array of basic features. It's designed to be lean and fast and will always stay that way.
  - Core should provide features that 80% or more of end users will actually appreciate and use.
- **Decisions, not Options**

# WordPress Release Cycle - Feature Plugins

- New features are no longer developed directly in core.
- Features are now developed as focused plugins.
- When the plugin is deemed “ready” it is submitted via a proposal to the core team for review.
- When approved by core committers, the code is revised to fit into WordPress core and committed.



&lt;&gt; Code

! Issues 59

🔗 Pull requests 16

📁 Projects 4

📶 Pulse

📊 Graphs

WP REST API - a JSON-based REST API for WordPress. <http://v2.wp-api.org>

🕒 3,948 commits

🌿 31 branches

🏷 54 releases

👤 88 contributors

Branch: develop ▼

New pull request

Create new file

Upload files

 **rachelbaker** committed on **GitHub** Merge pull request [#2662](#) from BE-Webdesign/[#1612](#)-comment-children-links

📁 assets

thicker text size on icons

📁 bin

"WP API" -&gt; "WordPress REST API" in README files

[Hide threads](#) | [Keyboard Shortcuts](#)[Reply](#) | [Follow](#)**Ryan McCue** 9:03 pm on September 20, 2015Tags: [merge \( 9 \)](#), [json-api \( 25 \)](#), [feature plugins \( 41 \)](#), [proposal \( 12 \)](#)

## WP REST API: Merge Proposal

Hello everyone! This is the post you've all been waiting for. 😊

We on the REST API team (myself, [@rachelbaker](#), [@joeHoyle](#), [@danielbachhuber](#), and newest member and core committer [@pento](#)) would like to propose merging the REST API into WordPress core. We've been working a while on this, and think it's now ready to get your feedback.

This is our first iteration of the proposal, and we're actively looking for feedback. If you have thoughts on the project, or on this proposal, let us know! Only with your feedback can we make progress. 😊

### What is the REST API?

# WordPress Release Cycle - Feature Plugins

- Started in June 2013 as a Google Summer of Code project.
- 3,948 commits and counting.
- 88 different code contributors.

# Tools



All WordPress development related conversation happens in Slack.

- Really great for tapping into the “heart beat” of the development process.
- Meetings happen here.
- General discussion about tasks.

# Tools



Let's check it out!

# Tools - Trac



The most important WordPress Core development tool.

Trac is the ticketing system that organizes all ongoing development in WordPress.

# Tools - Trac



Let's check it out!

# Tools - Local Machine

The following are tools you will need locally in order to contribute code to WordPress Core:

- Your favorite code editor.
- A local development environment (MAMP, XAMP, Vagrants, etc.)
- A working copy of the WordPress repository trunk branch
- A subversion client such as TortoiseSVN on PC or Cornerstone/Versions on Mac (if you are not comfortable with command line)



# Setting up our local development environment

Let's discuss what development stack we are going to set up today.

<https://make.wordpress.org/flow/2016/02/08/testing-patches-with-vvv-on-mac-os/>

# Setting up our local development environment

[virtualbox.org](https://www.virtualbox.org)



## VirtualBox

Allows an unmodified operating system with all of its installed software to run in a special environment, on top of your existing operating system. This environment, called a "virtual machine", is created by the virtualization software by intercepting access to certain hardware components and certain features.

# Setting up our local development environment

[vagrantup.com](https://vagrantup.com)

Vagrant is a tool for building complete development environments. With an easy-to-use workflow and focus on automation, Vagrant lowers development environment setup time, increases development/production parity, and makes the "works on my machine" excuse a relic of the past.



# Setting up our local development environment

[github.com/Varying-Vagrant-Vagrants/VVV](https://github.com/Varying-Vagrant-Vagrants/VVV)

The primary goal of Varying Vagrant Vagrants (VVV) is to provide an approachable development environment with a server configuration common for high traffic WordPress sites.

VVV is ideal for developing themes and plugins as well as for contributing to WordPress core.

# Setting up our local development environment



**VirtualBox**

[virtualbox.org](https://www.virtualbox.org)



VAGRANT

[vagrantup.com](https://www.vagrantup.com)



[putty.org](https://putty.org) (Windows users)

# Setting up our local development environment

1. Ensure Vagrant installed successfully.

```
Jonathans-MacBook-Pro:vvv Dayman$ vagrant -v  
Vagrant 1.7.4  
Jonathans-MacBook-Pro:vvv Dayman$
```

# Setting up our local development environment

2. Install the vagrant-hostsupdater plugin

```
Jonathans-MacBook-Pro:~ Dayman$ vagrant plugin install vagrant-hostsupdater
```

# Setting up our local development environment

3. Install the vagrant-triggers plugin

```
Jonathans-MacBook-Pro:~ Dayman$ vagrant plugin install vagrant-triggers
```



# Setting up our local development environment

4. Change directory to where you want the vagrant folder to reside.
5. Clone the VVV repo.

```
Jonathans-MacBook-Pro:~ Dayman$ git clone git://github.com/Varying-Vagrant-Vagrants/VVV.git vagrant-local
```

# Setting up our local development environment

6. Change directory to vagrant-local

```
Jonathans-MacBook-Pro:vvv Dayman$ cd vagrant-local
```

# Setting up our local development environment

## 7. Start the vagrant

```
Jonathans-MacBook-Pro:vagrant-local Dayman$ vagrant up
```

# Setting up our local development environment

## 8. Test it!

- <http://local.wordpress.dev/> for WordPress stable
- <http://local.wordpress-trunk.dev/> for WordPress trunk
- <http://src.wordpress-develop.dev/> for trunk WordPress development files
- <http://build.wordpress-develop.dev/> for the version of those development files built with Grunt
- <http://vvv.dev/> for a default dashboard containing several useful tools

# Next Steps

```
cd ~/Sandbox/vagrant-local/
```

```
vagrant up
```

```
vagrant ssh
```

```
cd /srv/www/wordpress-develop/
```

```
Jonathans-MacBook-Pro-2:vvv Dayman$ sudo npm install
```

# Types of Core Contribution we will cover

1. Testing Patches
2. Creating Patches

# Testing Patches

1. Read the ticket and all interaction referencing it thoroughly
  - Gain full understanding of what the ticket is trying to accomplish.
2. Look at when the patch was last created/refreshed.
  - Patches older than 3-4 months may need to be refreshed.
3. Review the patch.
  - Understand what is being changed so you know what to look for.

# Testing Patches

Provide feedback on the ticket.

- Does the patch apply correctly?
- Is the ticket solved with the given patch?
- Are there any bugs or edge cases that you can discover?
- Can you think of any negative effects as a result of the patch?
- Is this the best approach to solving the problem?



# Testing Patches

```
cd ~/Sandbox/vagrant-local/
```

```
vagrant up
```

```
vagrant ssh
```

```
cd /srv/www/wordpress-develop/
```

```
grunt watch &
```

```
svn up
```

# Creating Patches

- Is the ticket still valid? Old tickets can become
- Is there a clear direction of what needs to be done?
- Is there any discussion about how to approach the ticket? Sometimes there is an agreed upon solution without a patch.