



Setup

git

<http://git-scm.com/downloads>

ruby gems

package manager for ruby libraries

rails

gem install rails

rails new

create a new rails project



<https://s-media-cache-ak0.pinimg.com/originals/c4/34/84/c43484d186183ef943761926c779fef4.jpg>

Rake

ruby make

Rake Task

command line utility run with rake

rake --tasks

view all tasks available in the current context

rake about

details about the application environment

rake routes

display the routes defined in config/routes.rb



<http://i.ytimg.com/vi/O4pvFkRr8pI/maxresdefault.jpg>

Custom Task

use rake to automate anything you frequently do in the terminal



```
desc "I am short, but comprehensive description for my cool task"
task task_name: [:prerequisite_task, :another_task_we_depend_on]
do
  # All your magic here
  # Any valid Ruby code is allowed
end
```

http://guides.rubyonrails.org/command_line.html#custom-rake-tasks

Dependencies

tasks to run first

```
task :task_name, [:arg_1] => [:pre_1, :pre_2] do
  # You can use args from here
end
```

http://guides.rubyonrails.org/command_line.html#custom-rake-tasks

:environment

the task that sets up the rails context

Arguments



```
task :task_name, [:arg_1] => [:pre_1, :pre_2] do |t, args|
  # You can use args from here
end
```

http://guides.rubyonrails.org/command_line.html#custom-rake-tasks

Namespacing

group like tasks



```
namespace :db do
  desc "This task does nothing"
  task :nothing do
    # Seriously, nothing
  end
end
```

http://guides.rubyonrails.org/command_line.html#custom-rake-tasks

Try Me

write the following rake tasks:

- * repeat, takes and argument and echoes the argument
- * count:down which takes an integer and counts down one number per second until it gets to one



<http://imgs.abduzeedo.com/files/articles/baby-animals/Baby-Animals-042.jpg>

Workflow

the conventional practices of a person or team
with regards to modifying code and incorporating
those modifications

Your Own Git Repo

branch, add feature (and passing tests), merge into master

Sharing a Repo

branch, add feature (and passing tests), PR to master, receive comments on PR, make updates, get merged (hopefully)

branching

for every feature

committing

every. time. something. new. works
(or gets better)

... and sometimes for backups

pull requests

a request to incorporate changes from another branch
(or another fork)

code reviews

- another pair of eyes on changes.
- team member comments, suggests alternatives.
- some companies require changes to "pass a code review"
- before they're put in master

merging

incorporating a branch into another branch (most often,
into the master branch)

merge conflict

when git can't decide what code to keep

happens when master has changed since the
last time you branched

fork repo

one that was copied from an already existing repository

how might this workflow be different?

Getting Help

(from me)

push your code up to a branch

send me a link (via email) to the line you're confused about

describe the symptoms (with a screenshot)

Pair Programming

Two programmers, one text editor.

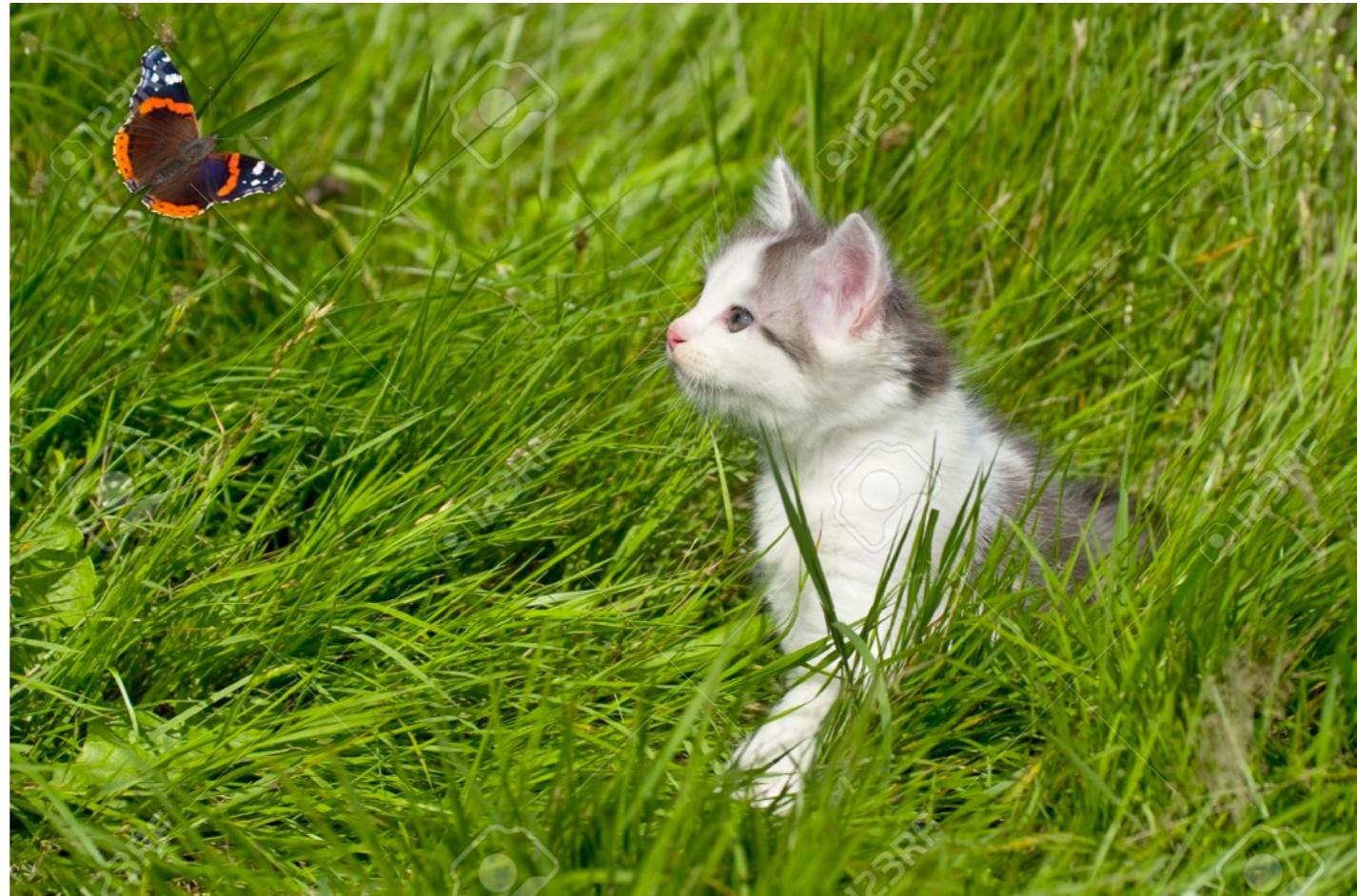


(is everyone in the class in seattle?)

Why?

Fewer Bugs

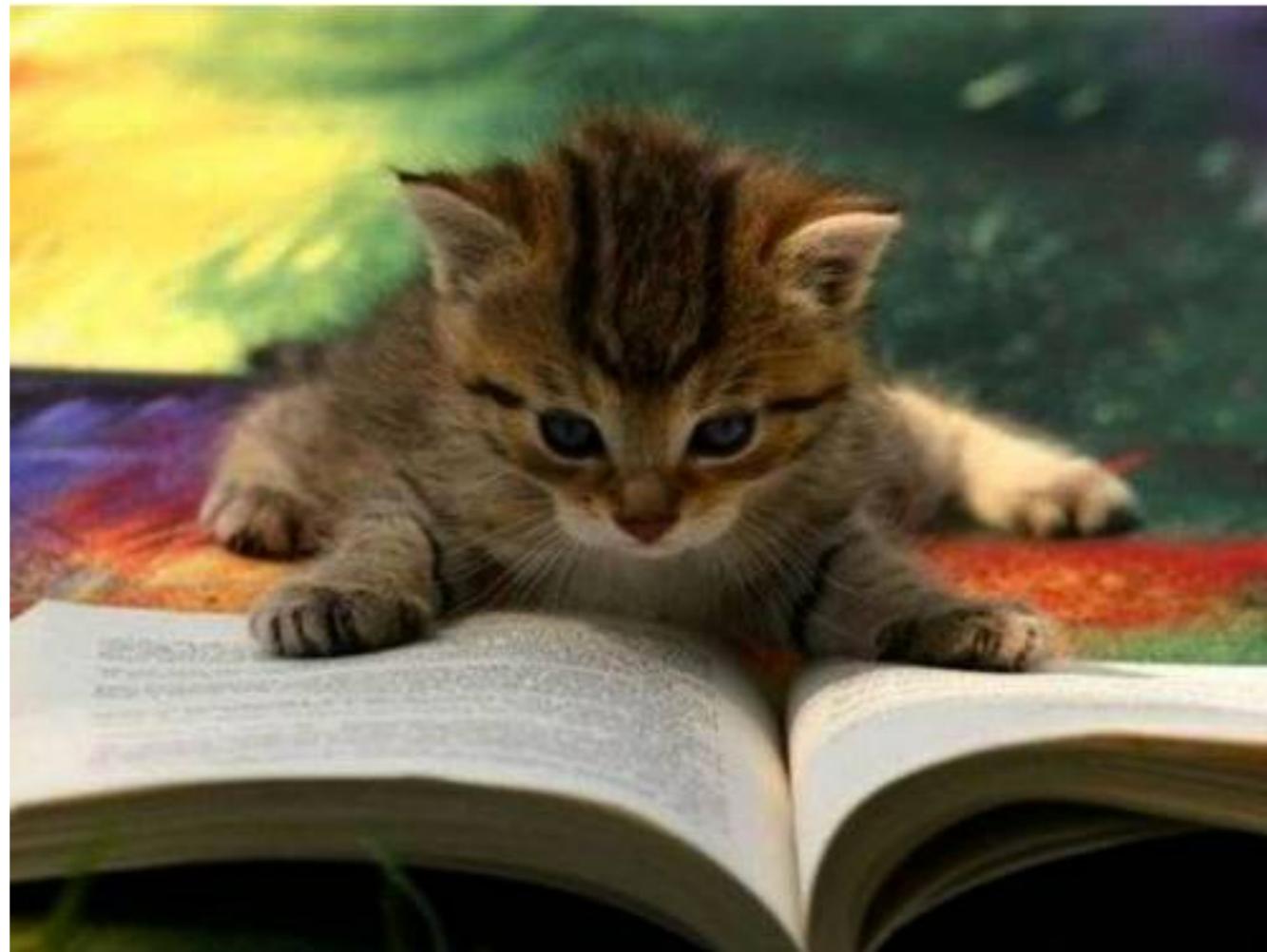
15% fewer bugs according to this study.



Learning

Share knowledge.

Explain your reasoning and test your assumptions.



Happiness

Makes coding more social. Helps you get unstuck.



Roles

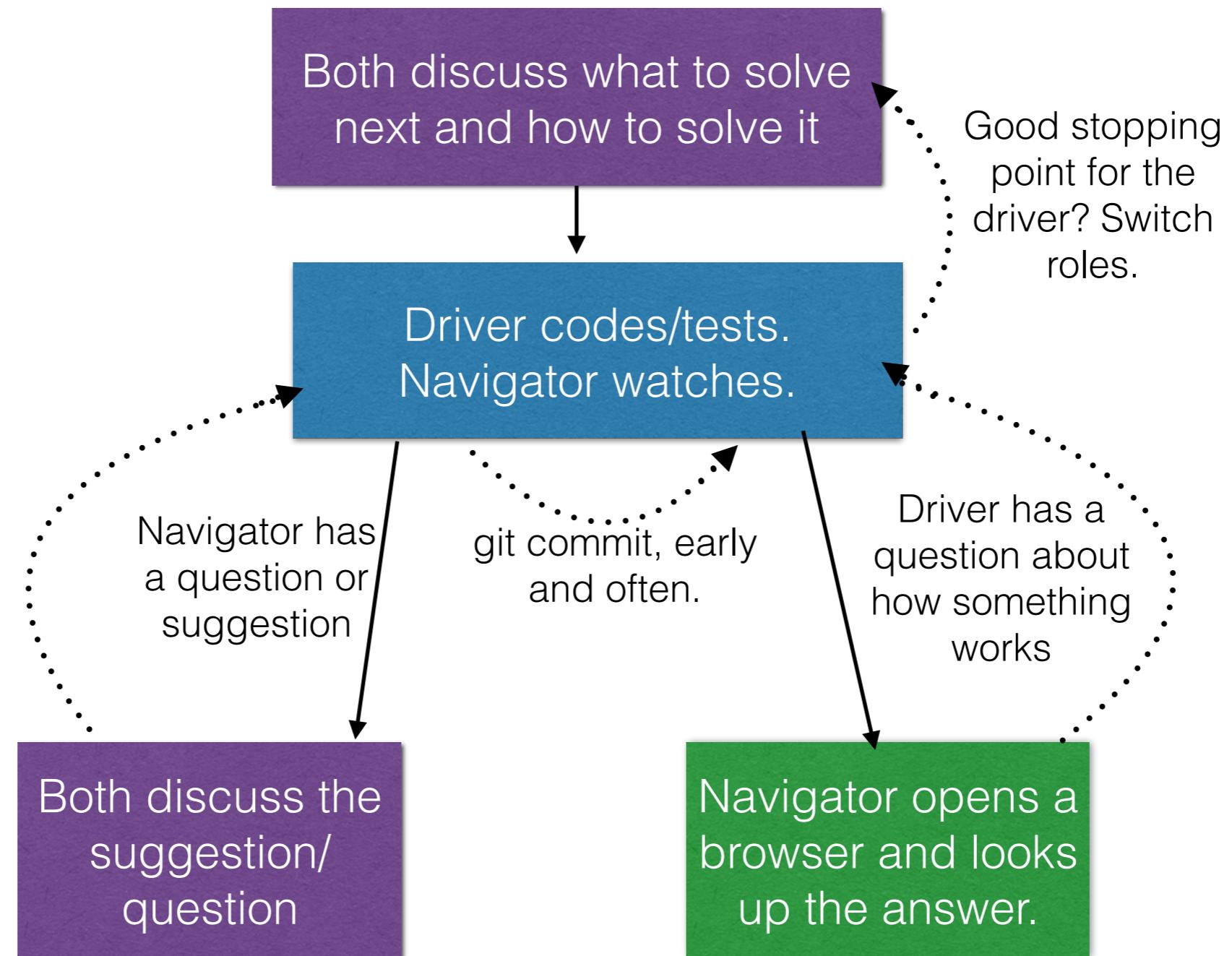
Driver

Write the code. Test the code.

Navigator

Read the code as the driver writes it,
make suggestions, ask questions.

Workflow



Save Your Changes

To make a git commit of all your changes so far, in the terminal:

```
$ git add .
$ git commit -m "this awesome thing we just did"
$ git push origin whateverBranchYoureOn
```

Switch Roles

The new driver needs to pull down the latest changes

```
$ git pull origin whateverBranchYou'reOn
```

Constructive Criticism

Be kind. Be generous. Be patient.



<http://maxcdn.thedesigninspiration.com/wp-content/uploads/2009/09/cute-animals/baby01.jpg>

Framework

A solution to the common parts of a frequent problem.
(like a mental framework)

MVC

web framework paradigm

Models

Interface to the database
Abstractions for business logic
Aggregation

Views

in general: the presentation layer

in rails: templates for creating http responses

Controller

data layer between the view and model

screenshots from:

http://guides.rubyonrails.org/action_controller_overview.html

Convention over Configuration

learn the magic. type less.



<http://justsomething.co/wp-content/uploads/2013/11/cutest-baby-animals-21.jpg>

Text Editors

pick one. learn it.

Sublime

<http://www.sublimetext.com/>

Atom

<https://atom.io/>



http://www.dvbsupport.net/images/justsmile/dsn64_large.jpg

Homework

- <http://docs.railsbridge.org/installfest/> (if you couldn't get rails new to work during class)
- <http://docs.railsbridge.org/intro-to-rails/> (including the extra credit)
- <http://docs.railsbridge.org/job-board/> download atom or sublime. do a tutorial: <https://www.shortcutfoo.com/app/dojos/sublime-text-2-mac/learn> or <https://atom.io/docs/v0.189.0/getting-started-atom-basics> or one of your choice.

email me if you have questions