

How much should I refactor?

@mattwynne
Software Archeology 2014











Refactoring:

If in doubt...

you're probably not
doing enough







Test-first

Greenfield

Small
team

No
commercial
pressure

Small,
frequent
commits

Regular,
diligent
refactoring



How many commits?

```
➤ git rev-list --all | wc -l  
160
```



160

Summer 2013



Assumption 1:

If it doesn't touch the
tests, it must be a
refactoring



➤ `git log --pretty=oneline --dirstat`

`82e9ac6b04395cec9bc635df424064e4606045bf` Sta

52.7% `lib/cucumber/core/ast/`

18.7% `lib/cucumber/core/gherkin/`

28.5% `spec/cucumber/core/ast/`

`7e19fa08d97b568c5deb13f29d5fcea75903933` Int

16.7% `lib/cucumber/core/test/`

5.0% `lib/cucumber/core/`

27.0% `spec/cucumber/core/test/`

43.0% `spec/cucumber/core/`

7.3% `spec/cucumber/`

58/160 commits
must be refactorings



It doesn't touch
the tests

36%



Flaw:

What about
renames?



Assumption 2:

If a commit message
says "refactor", that
commit is a
refactoring



A further 11/160
commits contained the
word **'refactoring'**



It doesn't touch
the tests

Commit message
contains "refactor"

43%



Flaw:

What about
renames?



Assumption 3:

If a the number of
tests doesn't change,
it must be a
refactoring



1. Iterate over the remaining commits
2. Check out the code
3. Run the tests
4. Check out the previous commit
5. Run the tests
6. Compare number of tests



it doesn't touch
the tests

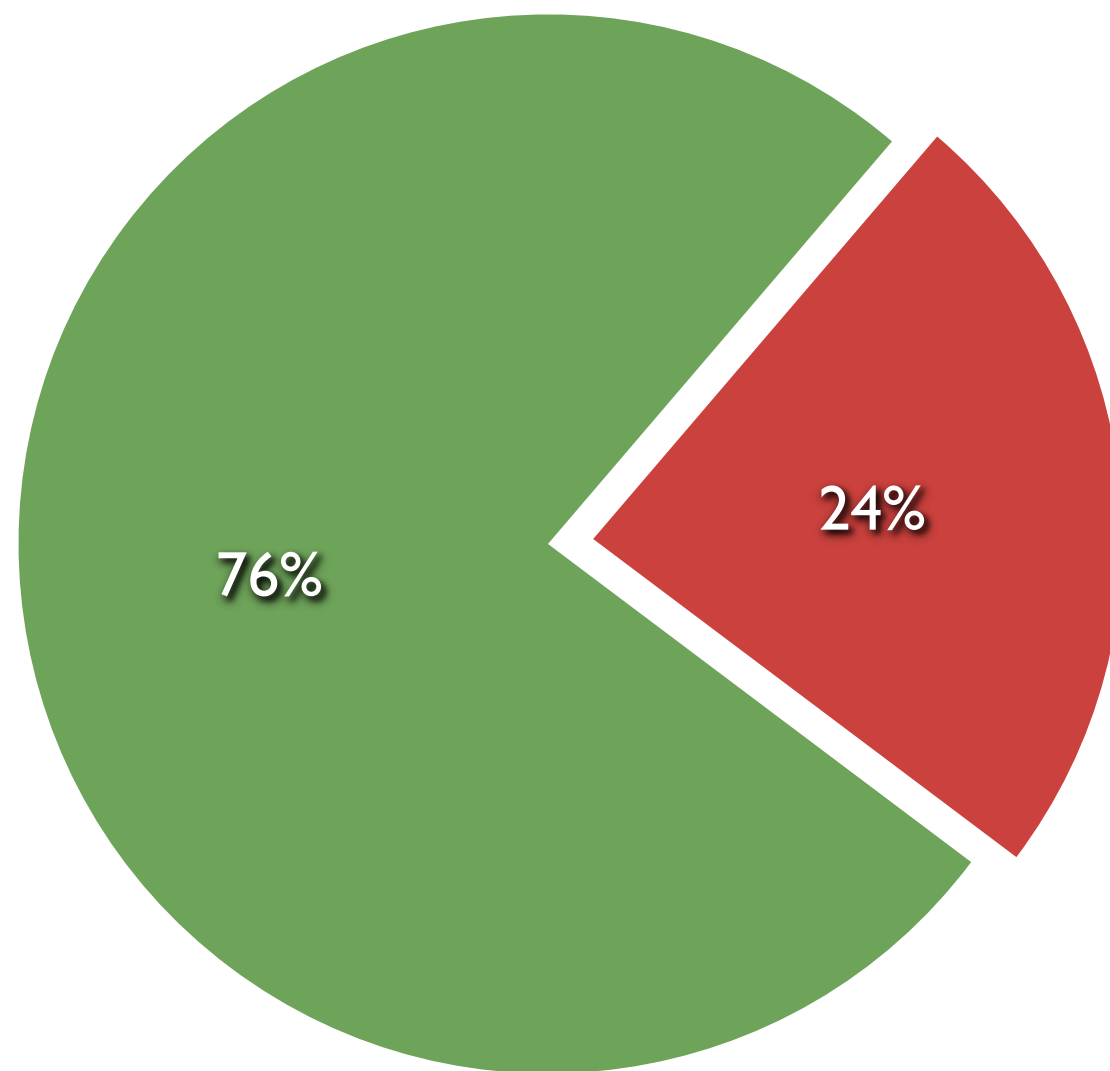
Commit message
contains "refactor"

76%

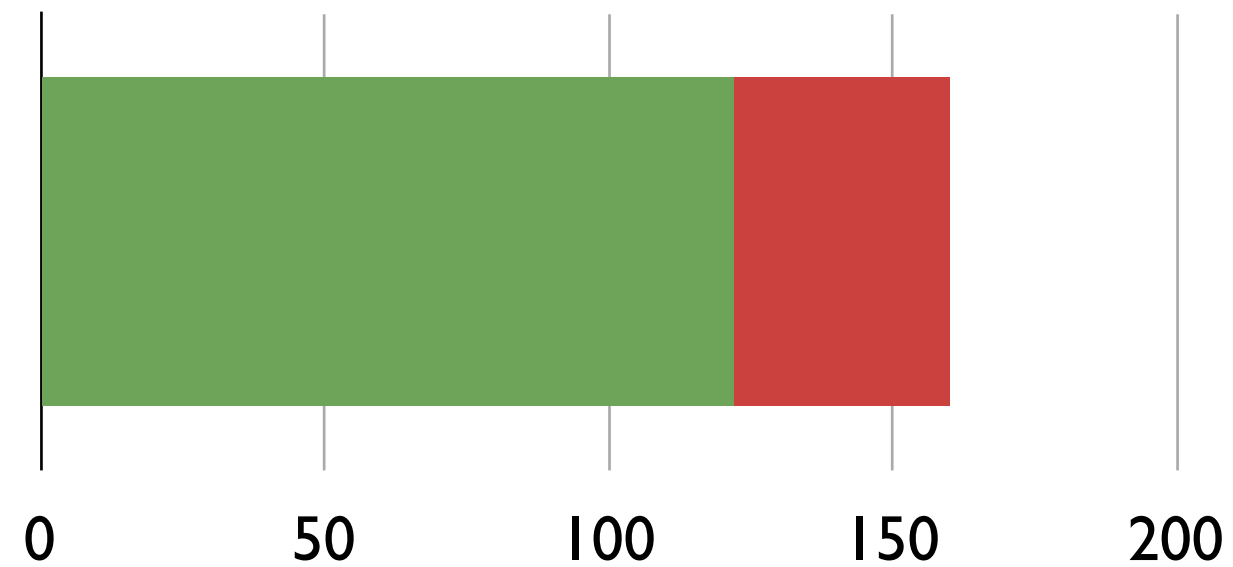
Number of tests
stays constant



● Refactoring ● New behaviour

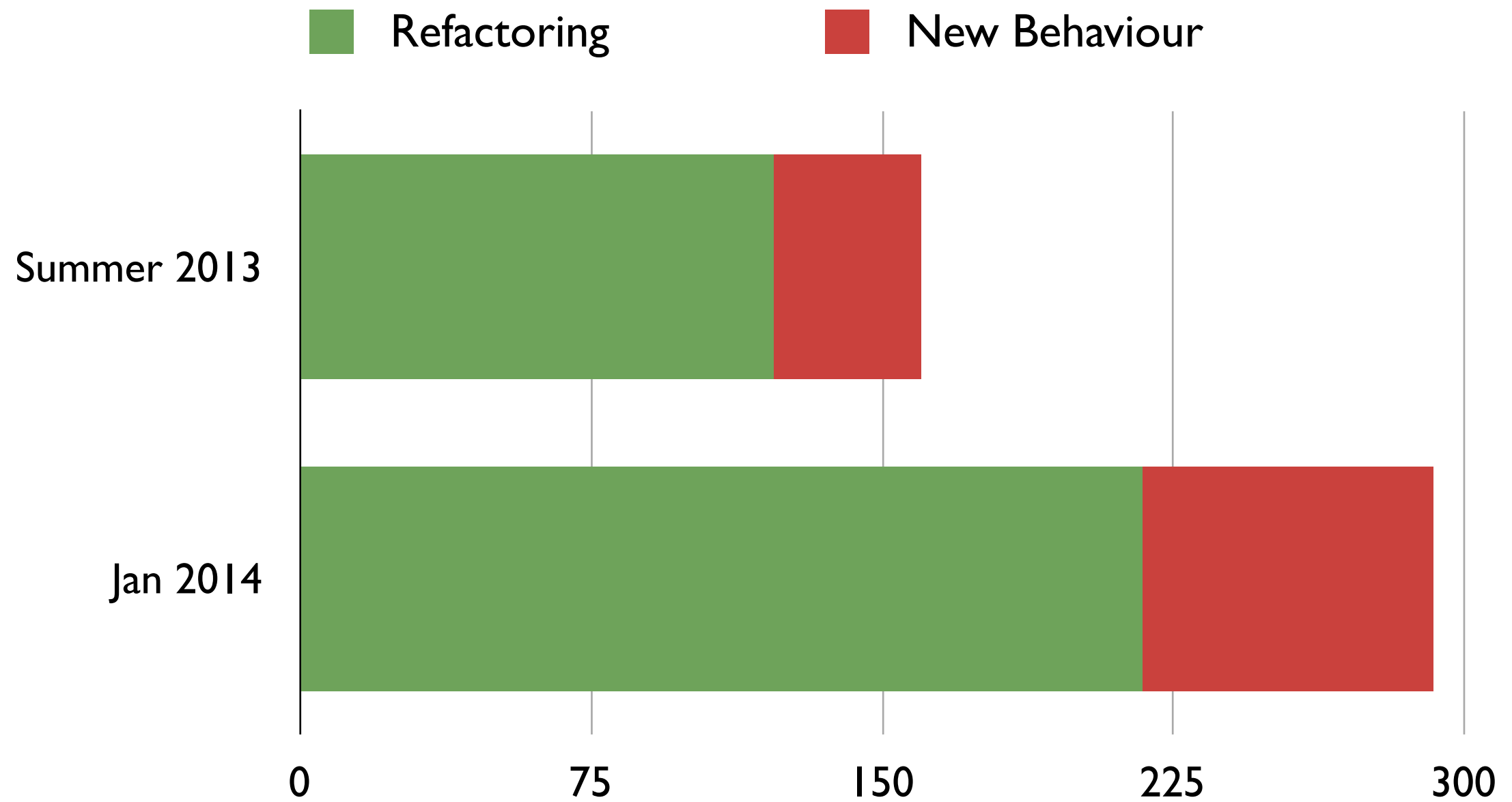


■ Refactoring ■ New behaviour



That was then...





Thanks :)

@mattwynne

<https://gist.github.com/mattwynne/6071100>

<https://github.com/cucumber/cucumber-ruby-core>

