Should testers participate in code reviews?

Seb Rose
Should testers participate in code reviews?

Seb Rose

TL;DR - Yes
Agenda

- Some definitions
- Primary motivation
- Typical approaches
- Two examples
- A technique
- Why testers?
- Over to you ….
- Wrap up
What is a review?

A **formal** assessment or examination of something with the possibility or intention of instituting change if necessary.
Formal

• done in accordance with rules of convention or etiquette
• suitable for or constituting an official or important situation or occasion
• officially sanctioned or recognised
What is a code review?

a formal assessment or examination of [something] with the possibility or intention of instituting change if necessary
What is a code review?

a formal assessment or examination of **software** with the possibility or intention of instituting change if necessary
What is a code review?

a formal assessment or examination of a **software change** with the possibility or intention of instituting change if necessary
What is a code review?

a formal assessment or examination of a **small software change** with the possibility or intention of instituting change if necessary
What is a peer review?

a formal assessment or examination of a small software change (by one or more of the author’s peers) with the possibility or intention of instituting change if necessary
Peer

a person that is of equal standing with another
Today’s working definition

a formal assessment or examination of a small software change (by one or more of the author’s peers) with the possibility or intention of instituting change if necessary
Why review?

@sebros

http://cucumber.io
Why review?

A. Style, structure, conformity
Why review?

A. Style, structure, conformity
B. Understandability
Why review?

A. Style, structure, conformity
B. Understandability
C. Performance efficiency
Why review?

A. Style, structure, conformity
B. Understandability
C. Performance efficiency
D. Bug prevention / detection
Why review?
The **PRIMARY** reason for carrying out a review is:

A. Style, structure, conformity  
B. Understandability  
C. Performance efficiency  
D. Bug prevention / detection
Why review?

The **PRIMARY** reason for carrying out a review is:

A. Style, structure, conformity
B. Understandability
C. Performance efficiency
D. Bug prevention / detection
E. None of the above
Simulation

Team Size: 7
Cumulative team size: 11 ± 2 @ 1σ

LoC: 157 k ± 23 k @ 1σ
Author present: 70% ± 14% @ 1σ

@sebrose
http://cucumber.io

@robsmallshire
Simulation

Team Size: 21
Cumulative team size: 114 ± 9 @ 1σ

LoC: 1.8 M ± 0.08 M @ 1σ
Author present: 19% ± 4% @ 1σ

@sebrose
http://cucumber.io

@robsmallshire
Simulation

Most authors of your product quit way back when

The proportion of code written by current team

20% after 20 years
Why peer review?
Why peer review?

To ensure that the next person who works with the code will be able to do so safely.
This is not news!

Programs must be written for people to read, and only incidentally for machines to execute.

Harold Abelson, 1984
How do we conduct reviews?
How do we conduct reviews?

Behind closed doors
How do we conduct reviews?

In pairs
How do we conduct reviews?

On our own
Which is best?
Which is best?

To ensure future understandability, reviews should be conducted without the presence of the author(s).
Sample pull requests (PRs)
Sample pull requests (PRs)

Other source control software is available
What do you think of this PR?

[Core Bug Fix] Fixed runtime exit status for ambiguous scenarios #1342

https://github.com/cucumber/cucumber-jvm/pull/1342/commits
What do you think of this commit?

https://github.com/cucumber/cucumber-jvm/pull/1342/commits

@sebros
What do you think of this commit?

https://github.com/cucumber/cucumber-jvm/pull/1342/commits
What do you think of this commit?

https://github.com/cucumber/cucumber-jvm/pull/1342/commits

@sebrosse
@Test
public void non_strict_with_ambiguous_scenarios() {
    Runtime runtime = createNonStrictRuntime();

    runtime.getEventBus().send(testCaseFinishedWithStatus(Result.Type.AMBIGUOUS));

    assertEquals(0x1, runtime.exitStatus());
}

@Test
public void strict_with_ambiguous_scenarios() {
    Runtime runtime = createStrictRuntime();

    runtime.getEventBus().send(testCaseFinishedWithStatus(Result.Type.AMBIGUOUS));

    assertEquals(0x1, runtime.exitStatus());
}
What do you think of this PR?

Not printing skipped items in cr-report by default. #374

https://github.com/sixty-north/cosmic-ray/pull/374/commits
What do you think of this commit?

https://github.com/sixty-north/cosmic-ray/pull/374/commits
```python
@-24,6 +24,8 @@ def _print_item(work_item, full_report):
    ret_val.append("timeout: {:.3f} sec".format(data))
else:
    ret_val = []

elif work_item.worker_outcome in [WorkerOutcome.NORMAL, WorkerOutcome.EXCEPTION]:
    ret_val += data
```
ret_val.append("timeout: {:.3f} sec".format(data))

else:
    ret_val = []

+ elif work_item.worker_outcome == WorkerOutcome.SKIPPED and not full_report:
+     ret_val = []

elif work_item.worker_outcome in [WorkerOutcome.NORMAL,
                                  WorkerOutcome.EXCEPTION]:
    ret_val += data
What do you think of this commit?

Not printing skipped items in cr-report by default.

report-hides-skipped-by-default (#374)

abingham committed a day ago

```python
2 cosmic_ray/reporting.py

@@ -24,6 +24,8 @@ def _print_item(work_item, full_report):
     ret_val.append("timeout: {:.3f} sec".format(data)
```
How were they different?
How were they different?

The test(s) acted as documentation of the developer's intent.
Unit tests as documentation

So if you want to go fast, if you want to get done quickly, if you want your code to be easy to write, make it easy to read.

Robert Martin
Unit tests as documentation

```java
@Test
public void scale() {
    Date now = new Date();
    Calendar calBegin = Calendar.getInstance();
    calBegin.setTime(now);
    calBegin.add(Calendar.HOUR, -4);
    Date begin = calBegin.getTime();
    Period p = new Period(4);
    long delta = p.getBegin().getTime() - begin.getTime();
    Assert.assertTrue(p.getEnd().compareTo(now) >= 0);
    logger.trace(delta);
    Assert.assertTrue(delta < 10 && delta > -10);
    Assert.assertEquals(new Integer(4), new Integer(p.getScale()));
}
```
Unit tests as documentation

```java
@Test
public void scale() {
  Date now = new Date();

  Calendar calBegin = Calendar.getInstance();
  calBegin.setTime(now);
  calBegin.add(Calendar.HOUR, -4);
  Date begin = calBegin.getTime();
}
```
Unit tests as documentation

```java
@Test
public void smoker_requires_manual_referral()
{
    Customer customer = customerBuilder
        .standardSingleFemale()
        .smoker()
        .build();

    Referral referral =
        underwriting.process(customer);

    Assert.assertEquals(Referral.Manual, referral);
}
```
Unit tests as documentation

```java
@Test
down
public void smoker_requires_manual_referral()
{
    Customer customer = customerBuilder
        .standardSingleFemale()
        .smoker()
        .build();
}
```
@Test
def public void smoker_requires_manual_referral()
{
    Customer customer = customerBuilder
        .standardSingleFemale()
        .smoker()
        .build();
}
How were they different?
How were they different?

Intention revealing names are easier to understand.
Including non-developers ...

• Ensures *diverse* perspectives
• Brings the *beginner’s eye*
• Keeps the language *ubiquitous*
Including testers …

• Brings the test perspective
• Spots missing tests early
• Allows testers to identify risk
• Helps cross-skill
TESTED =

CHECKED + EXPLORED

http://www.slideshare.net/ehendrickson/the-thinking-tester-evolved

@sebrose

http://cucumber.io
Readable unit test names

• Give visibility
• Allow focus on risk
• Minimise duplication of effort
• Maximise time spent exploring
Design your own review process ..... 

At your tables, define a review process that you’d like to use at your place of work:
Design your own review process ..... 

At your tables, define a review process that you’d like to use at your place of work:

• Define it’s purpose(s)
Design your own review process ..... 

At your tables, define a review process that you’d like to use at your place of work:

• Define it’s purpose(s)
• Decide how it should be conducted
Design your own review process ..... 

At your tables, define a review process that you’d like to use at your place of work:
• Define it’s purpose(s)
• Decide how it should be conducted
• Describe who should be involved AND what their responsibilities should be
Design your own review process ..... 

At your tables, define a review process that you’d like to use at your place of work:

• Define it’s purpose(s)
• Decide how it should be conducted
• Describe who should be involved AND what their responsibilities should be
• Suggest changes to your Definition of Done
Health Warning
Health Warning

Unvalidated hypothesis
Takeaways
Takeaways

A. Primary purpose of review is understandability
Takeaways

A. Primary purpose of review is understandability
B. Drive review from the tests
Takeaways

A. Primary purpose of review is understandability
B. Drive review from the tests
C. Exclude the author(s) from the review
Takeaways

A. Primary purpose of review is understandability
B. Drive review from the tests
C. Exclude the author(s) from the review
D. Include non-developers in the review
Should testers participate in code reviews?
Should testers participate in code reviews?

Should non-developers participate in code reviews?
Should testers participate in code reviews? **Yes!**

Should non-developers participate in code reviews? **Yes!**
Questions?

Seb Rose

Twitter: @sebrose
Blog: http://cucumber.io/blog
E-mail: seb@cucumber.io