BLRTMM6

Testers in product development - Code review phase

Chetan Giridhar

(http://technobeans.com)

Vishal Kanaujia

(http://freethreads.wordpress.com)

Objective

- Break the myth – “testers need not or shouldn’t review developer’s code”
- Substantiate the merits of the idea, analyzes the flip side.
- Audiences to evaluate the thought…

What is Code Review

- According to Wikipedia, *"Code review is systematic examination (often as peer review) of computer source code intended to find and fix mistakes overlooked in the initial development phase, improving both the overall quality of software and the developers' skills."*
- Code Review helps in finding or rather preventing critical defects early in the cycle (thereby reducing the bug fixing costs)
- Improving developers' knowledge and provide exposure to the other components of the software system.

Importance of code review

- Better quality of code
  - Logic and semantic errors
  - Potential bugs, corner cases
  - Design suggestions
  - UT suggestions
- Enforced coding standards
- Helps in improving skills of fellow developers
- Better cross domain understanding

Let's review code together

Ex)

```
void foo(char* input) {
    printf("Input char is %s", *(input+2));

    int *p = malloc();
    int* B[10];

    for (I =0; I < 10; i++) {
        free(p);
        int* q = malloc();

        B[i] = q;
    }
}
```

Ex)

```
email = None
def parse_email():
    global email
    if email is None:
        import email, re
```

Ex)

```
Use SQLiteHandler;
Sql = SQLiteHandler();
Sql.connect();
Sql.execute();
……..
……..
Sql.connect();
Sql.execute();
Sql.commit();
Sql.close();
```

Whose game is it?

- Traditionally, code reviews are done by developers and they can best review it!
- Testers were not expected to code; Coding skills were ‘good to have’!
- Now, industry prefers test engineers competent in development skills too
  - leveraging development perspective in test
  - ensuring better designed automation frameworks
  - white-box testing to improve coverage

For: Involve testers in code reviews

- **Utilizing Dev and QA outlook**
  - Development teams work with the intent of building the system.
  - QA thinks of ways of breaking the system.
  - Developers may oversee scenarios, which a tester may uncover.
  - Developers often neglect or de-prioritize UT for their modules.

- **QA builds product knowledge**
  - QA is only aware of the functionalities to be tested from the requirement specification documents.
  - Code reviews will help QA bridge gaps between their perception of the requirement and the actual code implementation.
  - QA understands the product better which is imperative for testing the product.

- **Improved test documentation**
  - Code reviews help QA identify gaps in test documentation.
  - Improved test plan and test case design documents.
  - Improved test automation.
- **Reduced testing time and costs**
  - If QA were to participate in code reviews, they wouldn't have to wait for the *RTQA* build to understand the implementation.
  - Speeds up testing and reduces the overall product life cycle time.

- **Better triaging from QA**
  - QA understands the code changes.
  - QA would try to pinpoint a failure as they are aware of the implementation.
  - Saves a lot of developer time.
  - Improves Dev/QA productivity.

- **Involvement of QA in Root Cause Analysis (RCA)**
  - QA is cognizant of all the functionalities and the implementation.
  - QA could assist in performing root cause analysis of defects that have been reported in the field.
- **Aids in white-box testing**
  - Taking part in code review would definitely help testers in improving and adding more tests in white-box testing.

- **Improved quality of code-comments and enforcing coding guidelines**
  - The idea here is that developer egos can be used to boost the quality of code. ☺
  - Developers will include better comments so that they don’t have to answer too many questions from the reviewers.

- **Help for a newly set-up development team**
  - New development team is set up to work on a research project.
  - Suggestions from senior QA members given to Dev as formal/informal inputs make developers comfortable.
  - Code review meetings can be seen as an opportunity for such discussions.
- **Improving rapport between Dev and QA**
  - As code reviews are helpful for both Dev and QA, they would be beneficial for developing a healthy environment among the teams.

Against: Code review is not for testers!

- **Steep learning curve**
  - Imperative to evaluate the development skills of the QA team .
  - Steep learning curve and time involved for QA to gain development or white-box testing skills to understand and review code.
- **Extra effort required by the testing team**
  - Better planning to include additional responsibility in QA cycle.
  - Impacts QA deadlines and may lead to project deadline slippages.

- **Testing based on implementation rather than specification**
  - QA test the product based on implementation knowledge and not based on the specification.
  - Defeats the whole purpose of QA perspective!
  - QA may start justifying the behavior of the product based on product knowledge rather than validating against requirements.
- **Over-involvement of testers understanding complex code**
  - Testers may get too interested in understanding a complex code snippet.
  - Tester spends too much time appreciating the code and spends less time in testing.

- **Code complexity/simplicity decides the testing strategy**
  - Depending on the complexity of the code, QA may change their testing strategy.
  - Testing effort projected based on code-complexity; wrong assumptions.
  - Results in incorrect test planning.

To summarize

- Not only do reviews help QA in improving the product knowledge, but also errors found by the QA team during code reviews can be fixed earlier in the cycle.
- Code reviews also help in improving test documentation and give QA an opportunity to identify more areas in testing .
- Moreover, developers also benefit as they get a chance to get inputs on the products from experienced QA who have worked on similar technologies before.
- May look like an attractive proposition, but it should be evaluated. Does it suit my needs? Does it solve my purpose?

Conclusion

We would definitely advocate it!

It’s worth a try!

Remember: Time is required before it starts paying dividends..

Your thoughts? ☺

Previous references

Chetan has written this article and was first published at TestingExperience titled ‘Advocating Code Review for Testers’.

Have a look at it at

http://technobeans.com/articles

Appreciated by Alan Page