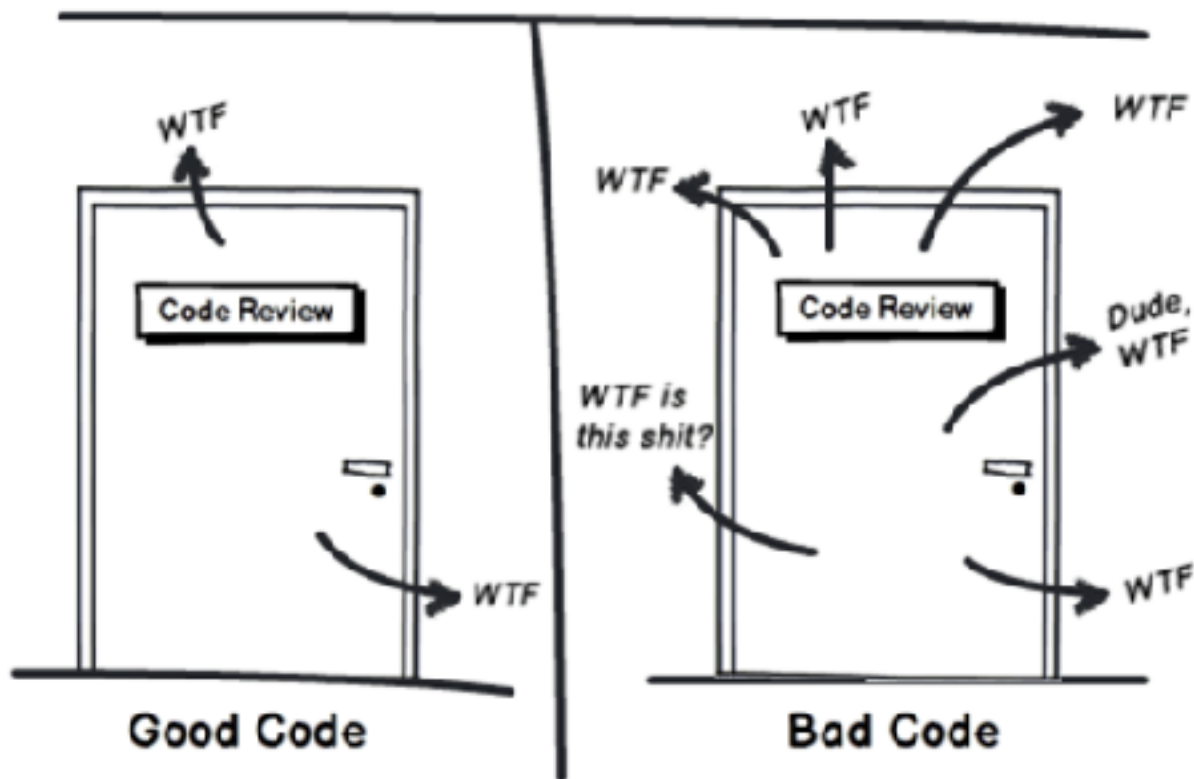


Code Quality Measurement: WTFs/Minute



Code Review

- “Code review is having other people look at your code in order to find defects.”

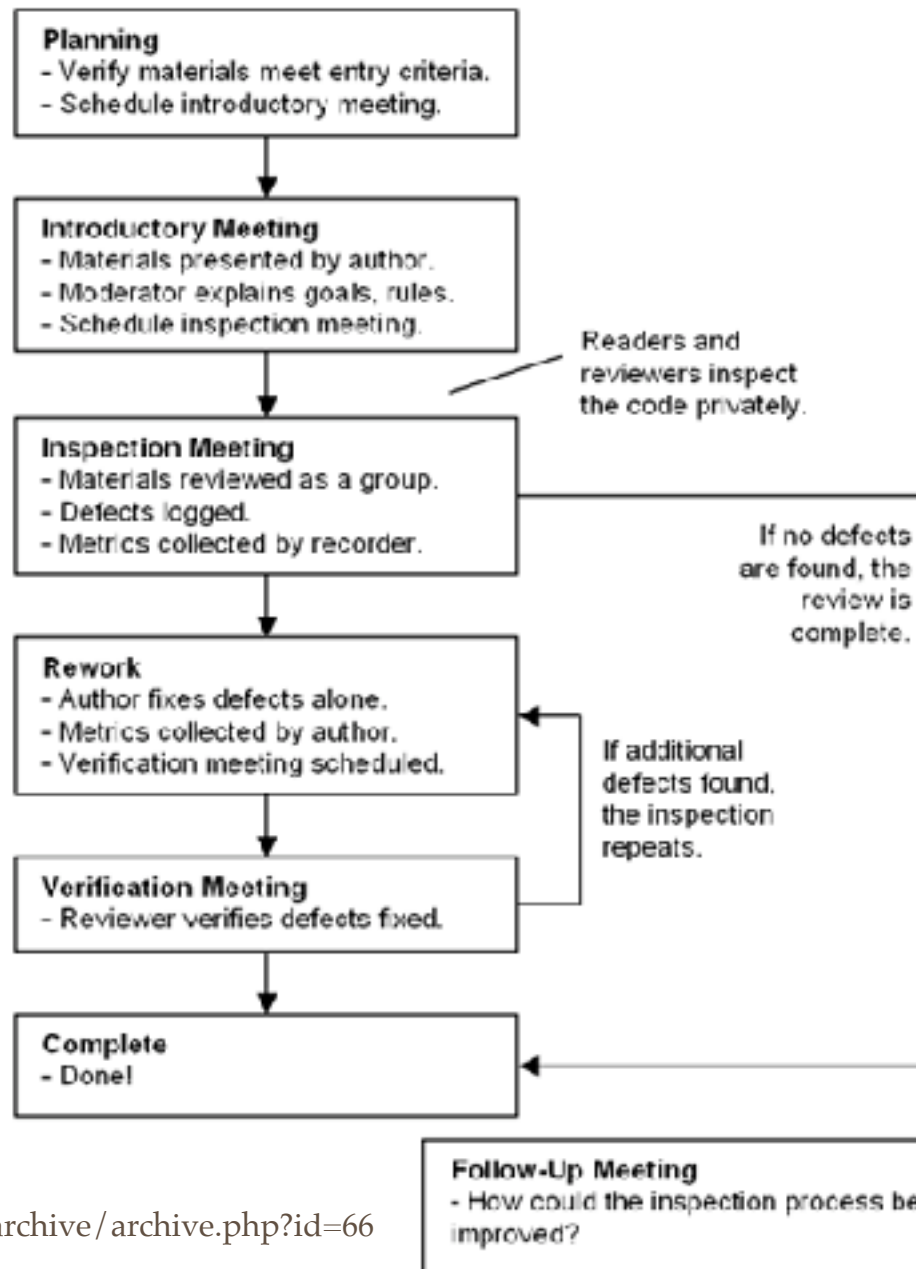
Code Review Pros and Cons

- + prevents releasing bugs
- + ensures architecture / code quality
- + leads to personal development
- takes time
- is impractical when reviewer doesn't know domain
- hurts feelings

Formal Inspection

- First developed by Michael Fagan in the mid 1970's.
- Very Specific Heavyweight process with 4 roles and 7 steps

A Typical Formal Inspection Process



Formal Inspection

- It Works, but is expensive.
- 9 person-hours per 200 lines of code
- Very impractical for today's realities

Light Weight Approaches

- Over the Shoulder
- Pair Programming
- Pull Requests

Over the Shoulder

- Reviewer sits with the developer and looks “over their shoulder” at the code.
- The reviewer can give informal feedback which can then be incorporated immediately if possible
- Heard about duck-debugging?



Over the Shoulder

- + Easy to Implement
- + Fast to Complete
- + Easy to quickly incorporate changes
- Reviewer cannot review at their own pace
- No Verification
- Reviewer only sees that developer shows them

Pair Programming

- Code is written by a pair, so Code Review is “Baked In” to the process

Pair Programming

- + Great for finding bugs and promoting knowledge transfer
- + Review is in-depth
- Reviewer is not objective
- Hard to do remotely
- No Verification

Pull Requests

- Code is peer reviewed as a part of the Pull Request process
- No pull request should be accepted without being reviewed **by a different developer**

Pull Requests Pros and Cons

- + Can be enforced by Version Control Practices
- + PR serves as verification of review
- + Can be done asynchronously
- + Reviewers can see all source code
- Might be hard to understand without explanation
- Most important changes can be lost with lots of small insignificant changes

Peer Review Best Practices: Architecture/Design

- Single Responsibility Principle
- Code Duplication
- Squint Test
- Leave Code Better
- Potential Bugs
- Error Handling
- Efficiency

Peer Review Best Practices: Style

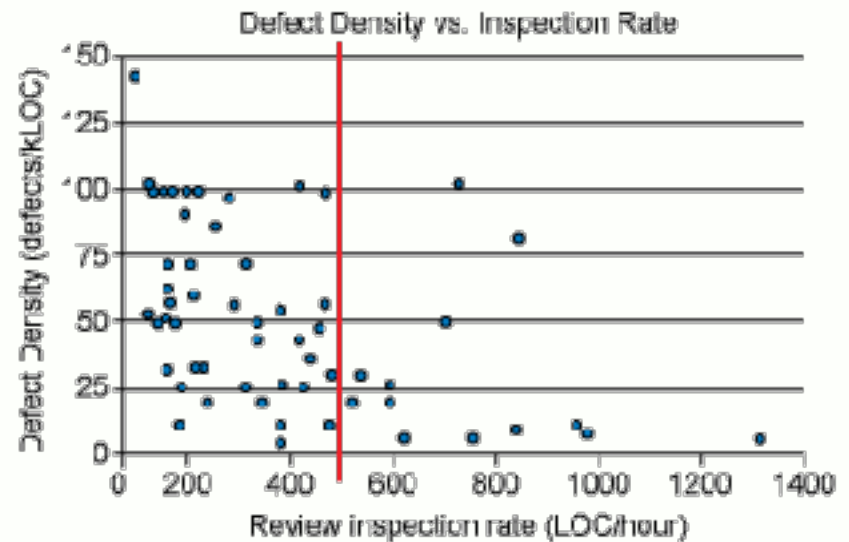
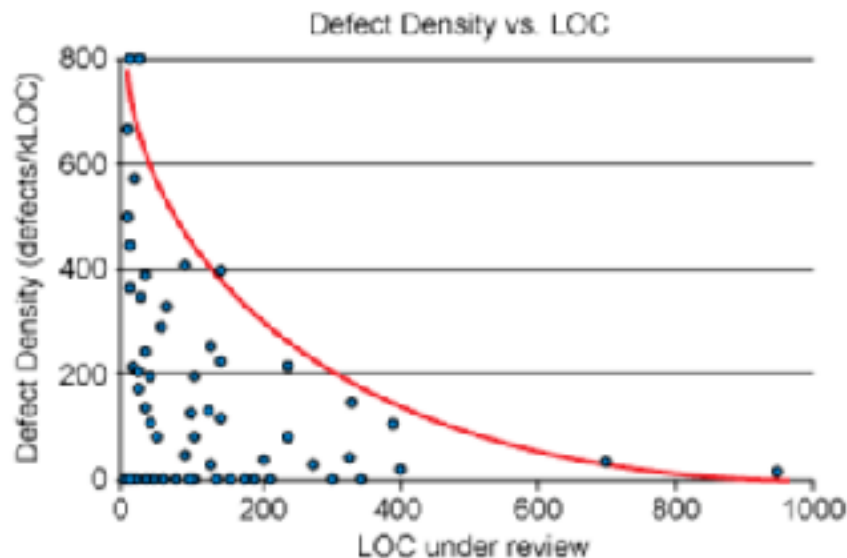
- Method Names
- Variable Names
- Function Length (~20 lines)
- Class Length (<100 lines)
- File Length
- Commented (out) Code
- Number of Method Arguments (<4)
- Readability (do I have to pause?)

Peer Review Best Practices: Testing

- Test Coverage
- Testing at the right level
- Meets requirements

Practical Suggestions

- Review < 400 LOC at a time
- Don't review > 60 min at a time
- Use a Peer Review Checklist (should be domain/language specific)
- Follow up with review comments
- Positive review culture



For Sprint 4

Go to a/the specific File changed in the PR

cs361fall2017 / sprint4

Unwatch 3

Star 0

Fork 0

Code

Issues 0

Pull requests 1

Insights

Settings

New dealAll method for dealing all cards #1

Edit

Open nelsonni wants to merge 1 commit into master from greatNewFeature

Conversation 0

Commits 1

Files changed 1

+9 -0



nelsonni commented 8 minutes ago

Owner



This PR adds a new `Game.java::dealAll()` method that allows the game to deal all cards instead of only dealing four at a time.



new dealAll method for dealing all cards

165ac1c

Reviewers



No reviews—request one

Assignees



No one—assign yourself

Labels



None yet

Add more commits by pushing to the `greatNewFeature` branch on `cs361fall2017/sprint4`.

Select the line you want to comment on

New dealAll method for dealing all cards #1

[Edit](#)[Open](#) nelsonni wants to merge 1 commit into `master` from `greatNewFeature`[Conversation](#) 0[Commits](#) 1[Files changed](#) 1

Changes from all commits ▾ Jump to... ▾ +9 -0

[Unified](#)[Split](#)[Review changes ▾](#)9 `src/main/java/models/Game.java`[View](#) @@ -22,6 +22,15 @@ public void dealFour() {

```
22 22      }
23 23      }
24 24
25 + public void dealAll() {
26 + +     while (true) {
27 +         for (int i = 0; i < 4; i++) {
28 +             Card c = deck.cards.get(0);
29 +             columns.get(i).cards.add(c);
30 +         }
31 +     }
32 + }
33 +
```

```
25 34 //customDeal to setup game for testing purposes (i.e. shuffled cards are random and hard to test)
26 35 public void customDeal(int c1, int c2, int c3, int c4) {
27 36     columns.get(0).cards.add(deck.cards.get(c1));
```

Write your “constructive” comment

New dealAll method for dealing all cards #1

[Edit](#)[Open](#) nelsonni wants to merge 1 commit into master from greatNewFeature[Conversation](#) 0[Commits](#) 1[Files changed](#) 1

Changes from all commits ▾ Jump to... ▾ +9 -0

Unified

[Split](#)[Review changes ▾](#)

9 ■■■■■ src/main/java/models/Game.java

[Show comments](#)[View](#)

```
@@ -22,6 +22,15 @@ public void dealFour() {  
22 22      }  
23 23      }  
24 24  
25 +   public void dealAll() {  
26 +   while (true) {
```

[Write](#)[Preview](#)

AA B I “ < > ↺ ⋮ ⋮ ⋮ ↶ ↷ @

This line causes an infinite loop. The 'dealAll' method never returns once called because of this loop.

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

Styling with Markdown is supported

[Cancel](#)[Add single comment](#)[Start a review](#)

```
27 +   for (int i = 0; i < 4; i++) {  
28 +       Card c = deck.cards.get(i);  
29 +       columns.get(i).cards.add(c);  
30 +   }
```

You can collect comments

Changes from all commits ▾ Jump to... ▾ +9 -0 ■■■■■

Unified

Split

Review changes 1 ▾

9 ■■■■■ src/main/java/models/Game.java

☒ Show comments

View



```
22 22      }  
23 23      }  
24 24  
25 + public void dealAll() {  
26 +     while (true) {
```



nelsonni

Owner

Pending



This line causes an infinite loop. The `dealAll` method never returns once called because of this loop.



Reply...

Start a new conversation

Finish your review

```
27 +     for (int i = 0; i < 4; i++) {  
28 +         Card c = deck.cards.get(0);  
29 +         columns.get(i).cards.add(c);  
30 +     }
```

Finalize your review

Changes from all commits ▾ Jump to... ▾ +9 -0

Unified

Split

Review changes 2 ▾



nelsonni

Owner

Pending

This line causes an infinite loop. The `dealAll` method never returns once called because of this loop.



Reply...

Start a new conversation

Finish your review

```
27 +         for (int i = 0; i < 4; i++) {  
28 +             Card c = deck.cards.get(0);
```



nelsonni

Owner

Pending

This code bypasses the `Deck.java:dealFour` method and instead directly accesses the cards within the `Deck` class. This is an instance of the Inappropriate Intimacy code smell (<https://refactoring.guru/smells/inappropriate-intimacy>). A possible solution would be to use the Move Method refactoring to create a new `dealCard` method in the `Deck` class.



Reply...

Start a new conversation

Finish your review

```
29 +             columns.get(i).cards.add(c);  
30 +         }
```

Finalize your review

Summarize all your
low level comments

Request change or
Comment based on
what you want

Approve

Unified Split Review changes 2

Submit your 2 pending comments

Review summary

These changes would cause our game to become unplayable, since the infinite loop would halt gameplay at the first call to the new dealAll method. I won't approve these changes until both issues are fixed.

☒ Comment
Submit general feedback without explicit approval.

☐ Approve
Submit feedback and approve merging these changes.

☐ Request changes
Submit feedback that must be addressed before merging.

Submit review

ectly ac
nell
olution would be to use the Move
iss.

Your Turn (in pairs)

- Open a pull request by Nick in Sprint 4
- Use the checklist to find problems
 - Method Names
 - Variable Names
 - Function Length (~20 lines)
 - Class Length (<100 lines)
 - File Length
 - Commented (out) Code
 - Number of Method Arguments (<4)
 - Readability (do I have to pause?)

Thursday

- No Clicker quiz on Thursday
 - Instead make it a Canvas quiz (Testing-Refactoring)
 - Extra credit
 - Due on Friday

Thank for being a great class!!!

