



Why Development Teams are Throwing Away Institutional Knowledge and How to Preserve it

Peter Pezaris, CEO
pez@codestream.com

23 Years Together



Background

- Serial Entrepreneur
- CEO of 4 startups in 23 years
- Expert in real-time communication
- Every-day developer

**THE PROBLEM is
COMMUNICATION
(or lack thereof)**

Source Code



Discussion

??????

Source Code



Discussion

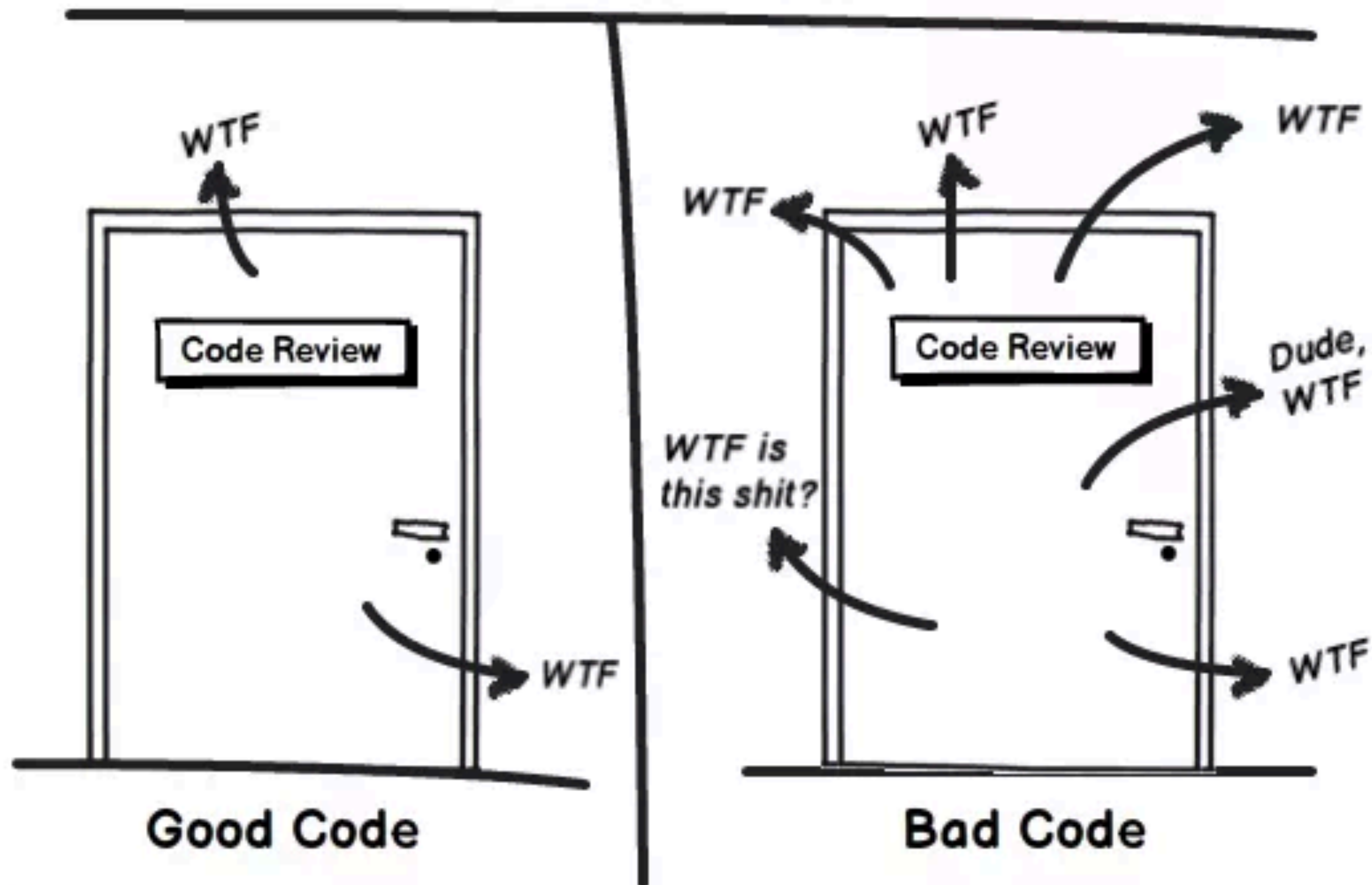


```
1 // utility math functions
2
3 function squared(x) {
4     return x * x;
5 }
6
7 function isPrime(num) {
8     if (num === 2) {
9         return true;
10    } else if (num > 1) {
11        for (var i = 2; i < num; i++) {
12            if (num % i !== 0) {
13                return true;
14            } else if (num === squared(i)) {
15                return false;
16            } else {
17                return false;
18            }
19        }
20    } else {
21        return false;
22    }
23 }
24
25 function isOdd(num) {
26     if (num === 2147483647) {
27         return true;
28     } else if (num === 0) {
29         return false;
30     }
31     return isOdd(num + 2);
32 }
33
```

WTF

- What am I looking at?
- How does this work?
- Why would we do it this way?

Code Quality Measurement: WTFs/Minute



The Bigger Problem

- Developers don't discuss code often enough
- When they do, it's not preserved for others to use or learn from

**WHY ARE CODE
DISCUSSIONS RARE?**

Reason #1

Reason #2

Reason #3

DEVELOPMENT process

- Agile or Waterfall Process
- Not designed to promote knowledge sharing
- Tasks assigned at start of sprint; little interaction until end of sprint

Developer Culture

Reason #1

Reason #2

Reason #3

- The Rockstar Coder
- “Don’t Ask Questions”
- Fear of looking stupid
- Really hard to change culture
- 40%: “competing with my peers”

WRONG OR MISSING TOOLS

Reason #1

- Screenshares
- Screenshots
- Copy/paste
- Slack
- Skype
- FB Messenger
- Whatsapp
- FaceTime
- Email
- Etc!

Reason #2

Reason #3

None of these tools know what code is!

<20%

Of Developers Learn Through
Internal Documentation

```

1  // utility math functions
2
3  function squared(x) {
4      return x * x;
5  }
6
7  function isPrime(num) {
8      if (num === 2) {
9          return true;
10     } else if (num > 1) {
11         for (var i = 2; i < num; i++) {
12             if (num % i !== 0) {
13                 return true;
14             } else if (num === squared(i)) {
15                 return false;
16             } else {
17                 return false;
18             }
19         }
20     } else {
21         return false;
22     }
23 }
24
25 function isOdd(num) {
26     if (num === 2147483647) {
27         return true;
28     } else if (num === 0) {
29         return false;
30     }
31     return isOdd(num + 2);
32 }
33

```

The Problem

- What am I looking at?
- How does this work?
- Why would we do it this way?

Inline = no conversation

Inline = no conversation

GitHub = limited & buried

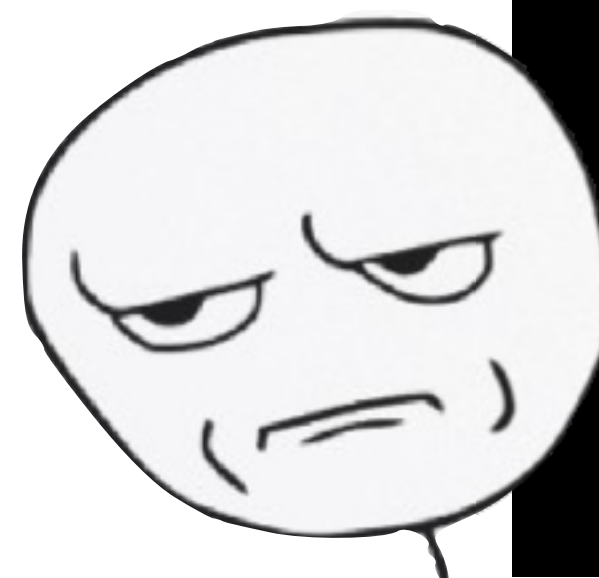
Inline = no conversation

GitHub = limited & buried

Slack = ephemeral

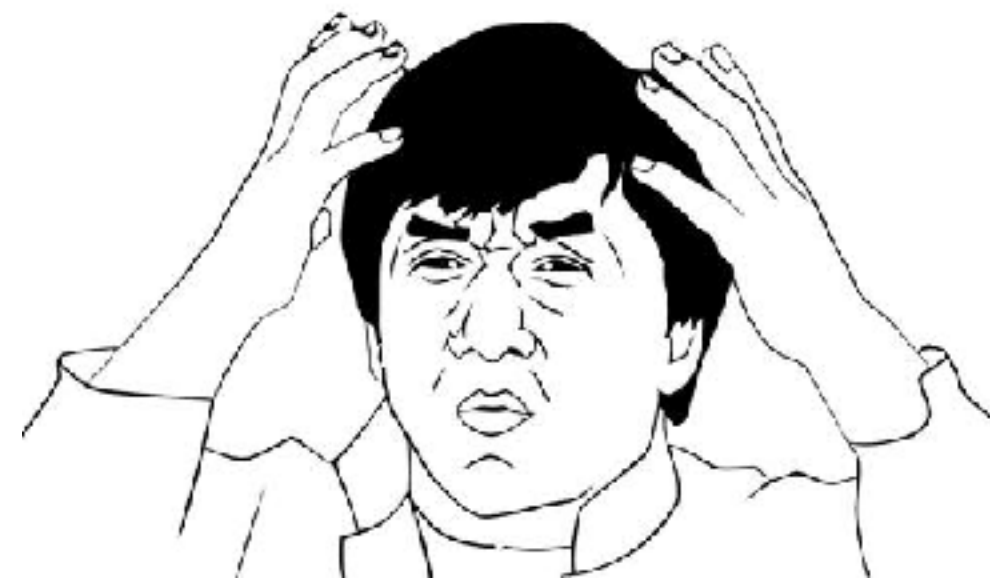
Slack

1. Select codeblock
2. Press Copy
3. Alt-tab to slack
4. Find and select the appropriate channel
5. Type ``` then paste then ``` again
6. Manually create context
 7. Enter project/repo details
 8. Type path and source file
 9. Copy line number
 10. Reference which function(s)
11. Find the right person to ask
 12. Alt-tab to Terminal
 13. cd to right directory
 14. git blame | grep
 15. Alt-tab back to Slack
 16. copy / paste author email to at-mention
17. Type your question (finally!)
18. Alt-tab back to your IDE



Slack

- Comments are lost in stream
- New devs never see them



**Why does KNOWLEDGE
RETENTION matter?**

TECH DEBT

ONBOARDING

TURNOVER

REMOTE

- Expensive
- Limiting
- Like Death and Taxes
- A constant struggle to maintain code quality

TECH DEBT

ONBOARDING

TURNOVER

REMOTE

- Adding developers to projects
- Reduce time to positive contribution
- Short-term drag on productivity

TECH DEBT

ONBOARDING

TURNOVER

REMOTE

- Tech turnover: 2x economy
- Median tenure: 2 years
- Losing Rock Star is most painful
- 10,000 Baby Boomers retiring every day

TECH DEBT

ONBOARDING

TURNOVER

REMOTE

- #1 reason to quit a job
- #1 and #3 perk
- 79% of knowledge workers do at least some work remotely



The Knowledge Base
Behind Your Codebase

1. Make it easy to talk about code.

Slack

1. Select codeblock
2. Press Copy
3. Alt-tab to slack
4. Find and select the appropriate channel
5. Type `` then paste then `` again
6. Manually create context
 7. Enter project/repo details
 8. Type path and source file
 9. Copy line number
 10. Reference which function(s)
11. Find the right person to ask
 12. Alt-tab to Terminal
 13. cd to right directory
 14. git blame | grep
 15. Alt-tab back to Slack
 16. copy / paste author email to at-mention
17. Type your question (finally!)
18. Alt-tab back to your IDE



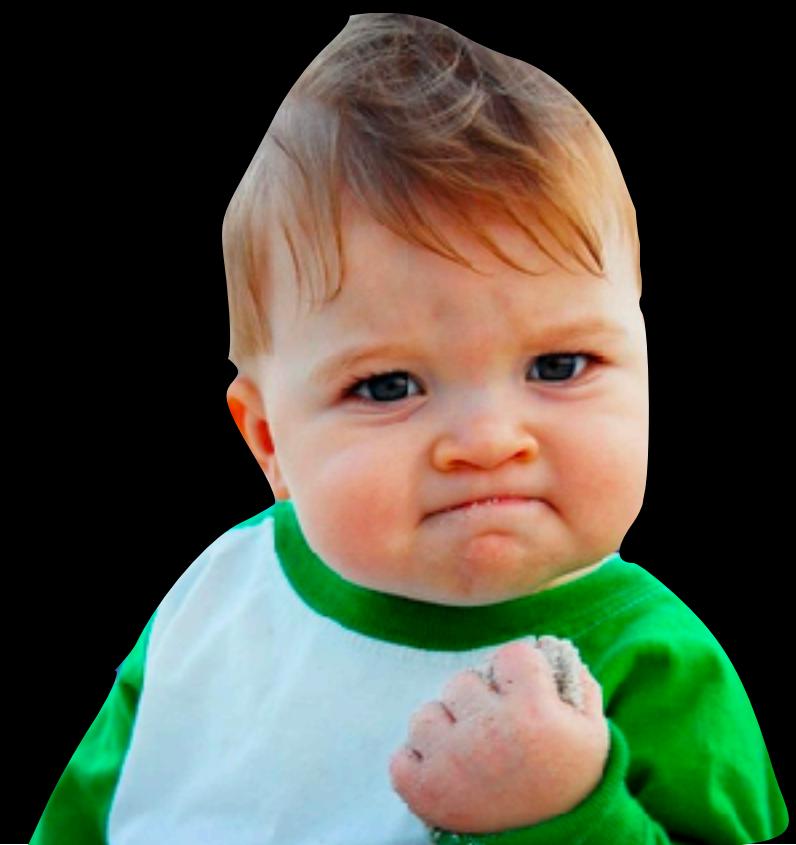
Slack

1. Select codeblock
2. Press Copy
3. Alt-tab to slack
4. Find and select the appropriate channel
5. Type ``` then paste then ``` again
6. Manually create context
 7. Enter project/repo details
 8. Type path and source file
 9. Copy line number
 10. Reference which function(s)
11. Find the right person to ask
 12. Alt-tab to Terminal
 13. cd to right directory
 14. git blame | grep
 15. Alt-tab back to Slack
 16. copy / paste author email to at-mention
17. Type your question (finally!)
18. Alt-tab back to your IDE



1. Select codeblock

2. Type question



2. Save the knowledge
forever

Slack

- Comments are lost in stream
- New devs never see them



Slack

- Comments are lost in stream
- New devs never see them



- Threads are saved with source file
- Knowledge base forms over time



DEMO



The Knowledge Base
Behind Your Codebase

GRAVEYARD

Source Code



Discussion

