



By

Annabelle, Ethan, Keegan, Tim

Problem

Team communication when using traditional code review tools hinders the development of **conversation** through **slow methods** of exchanging messages.

- This hinders conversation and leads to wasted programmer hours from inefficiency.
- This is a problem for which the solution is simple due to the existence of communication tools such as Slack

Current Approach

GitHub comment feature often results in days between messages



Our Solution

A Slack Plug-In for rapid code review communication



Q Review 12:18 PM

Hi 🖐️

You have a new pull request awaiting review from @Keegan

Here is the snippet

commit 404696c ▼

```
1 174 174 | int x = 0
2 175 175 |
3 176 176 | for int i in range(1, 10):
4 176     + print i
5 176     - print "hello world"
```

GUI Prototype

GUI Prototype

Comments



Annabelle Zha 1:33 PM

/qreview comment 178+ "Is this line also for debugging?"



Q Review 2:02 PM

I've added a comment to line 178+ 

GUI Prototype

Changes



Ethan Cui 1:21 PM



/qreview request changes “please remove the debug code in our production build”



Q Review 1:22 PM

Changes requested for commit `404696c`  [@Keegan](#)

GUI Prototype

Changes



Q Review 1:24 PM

Changes were added to this pull request by [@Keegan](#)

commit 4046821 ▼

```
1 174 174 | int x = 0
2 175 175 |
3 176 176 | for int i in range(1, 10):
4 176 -   print "hello world"
5 177 177 |     if (i % 2 == 0):
6 178 +       print "even number"
7 178 -       print "odd number"
```


GUI Prototype

And Most Importantly, Shipping Code!



Annabelle Zha 2:44 PM

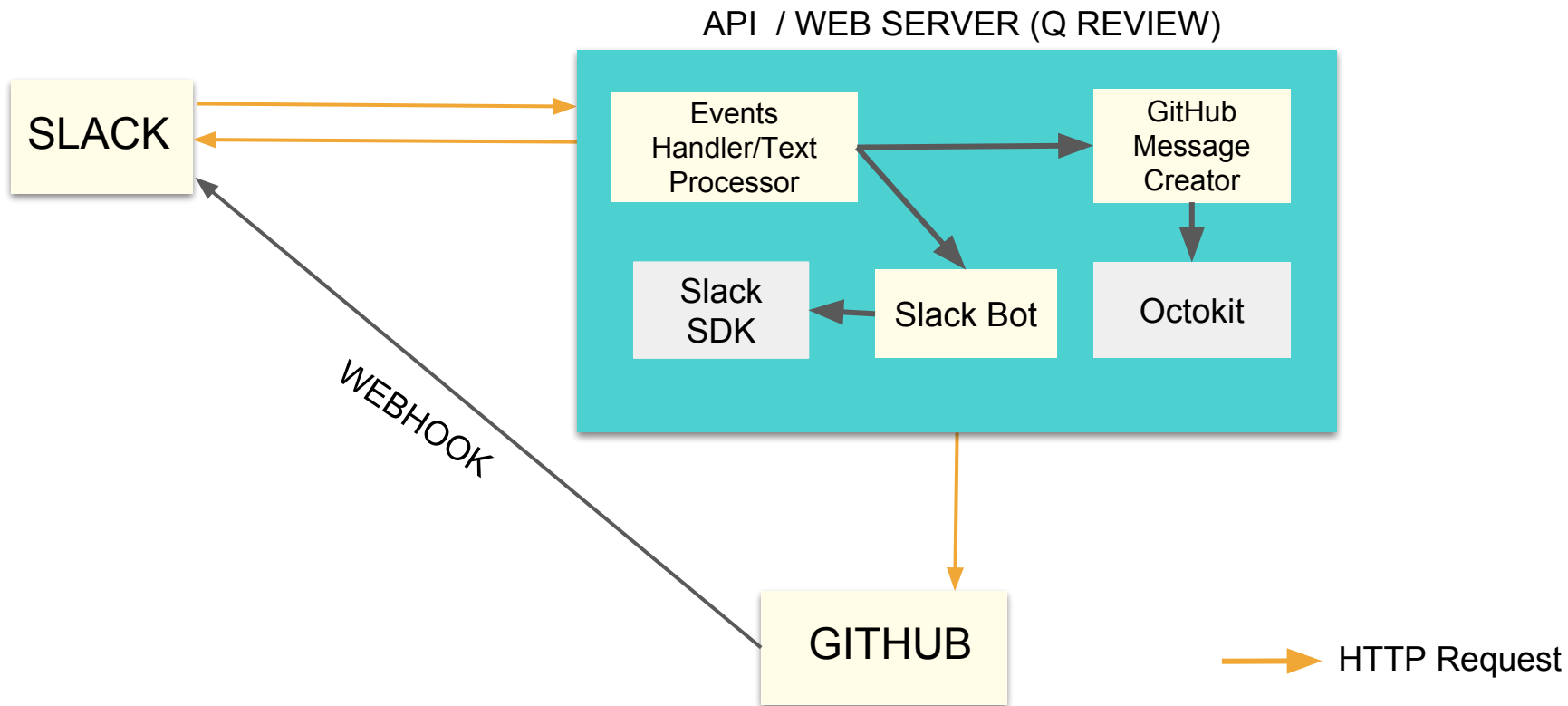
/qreview approve "Good Job! Thanks for getting this done!"



Q Review 2:44 PM

Great! I'm approving these changes 🚀

Our Architecture



Our Differences

Real-Time Communication

- Makes highly organized code review process

Based On Slack, the most popular workplace communication platform

- Avoids going back and forth between github and email inbox

Research Questions

Hypothesis

- Q Review increases the productivity of collaborative code review

Methodology

- Run an experiment on a random pull request to be reviewed, compare time spent to complete code review using traditional methods & Q Review

Information & Metrics

- Time spent on each part of the review process and total time spent

Real-world Relation

- Mimic the workflow of a code review but less friction than traditional approach

Validity Threat

- Proficiency in technology mismatch (between us and users)
- Unrepresentative pull request

Preliminary results

Lessons Learned:

- Verify assumptions early to redesign approaches to the solution
- What we think is clear may not be clear to the customers

Challenges:

- Retraining users who are accustomed to current approaches
- Working with available functionality in major tool APIs

What We Wish We Had Known:

- Experience with web servers and the Node.js/Express stack

Related Works

Atlassian Crucible

Our Difference:

- Free
- Based on Slack
- Expandable to multiple code review tools

