

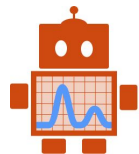
# THE ART OF GIVING AND RECEIVING **CODE** **REVIEWS**

Dr Alex Hill, Imperial College London

[alex.hill@gmail.com](mailto:alex.hill@gmail.com)

[github.com/hillalex](https://github.com/hillalex)

[@alexhillphd](#)



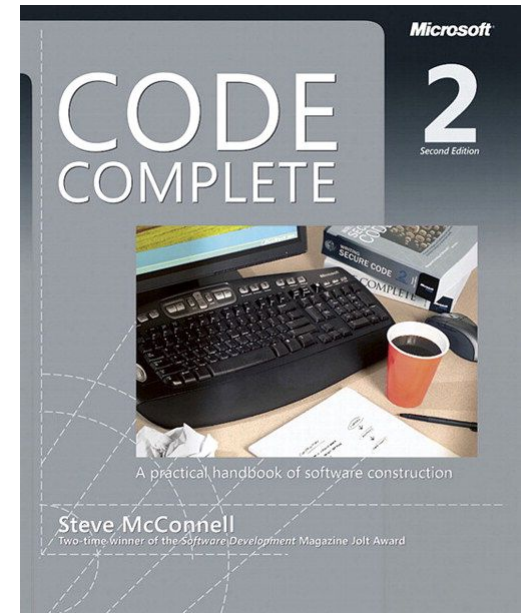
**RESIDE**

**MRC**

Centre for  
Global Infectious  
Disease Analysis

# 1. Defect finding

- In a software-maintenance organization, **55%** of one-line maintenance changes were in error before code reviews were introduced. After reviews were introduced, only **2%** of the changes were in error.
- In a group of 11 programs developed by the same group of people, the first 5 were developed without reviews. The remaining 6 were developed with reviews. After all the programs were released to production, the first 5 had an average of **4.5 errors per 100 lines** of code. The 6 that had been inspected had an average of only **0.82 errors per 100 lines**.
- A study of an organization at AT&T with more than 200 people reported a **90% decrease in defects** after the organization introduced reviews.



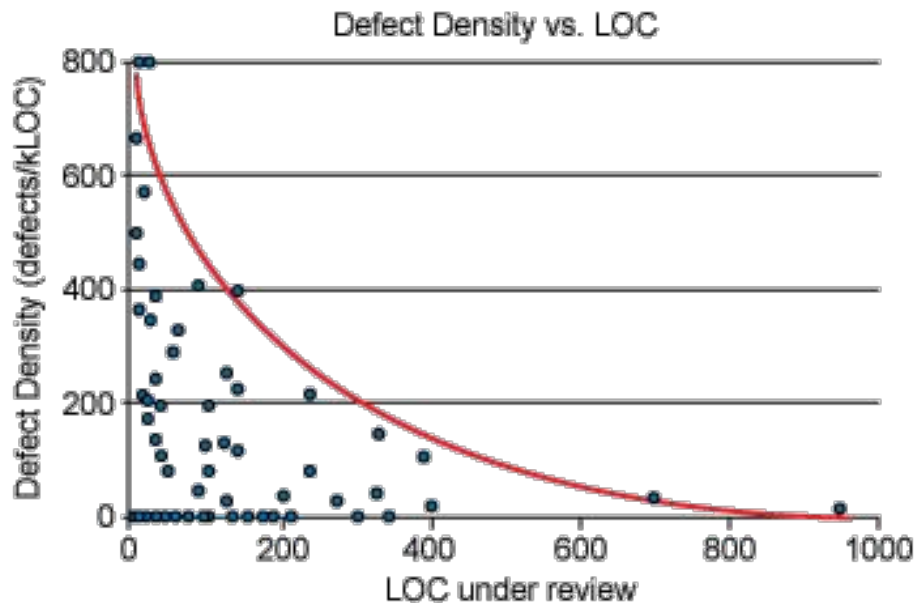
**Especially useful where automated tests are of limited use!**

## 2. Learning opportunities

## 3. Sustainability:

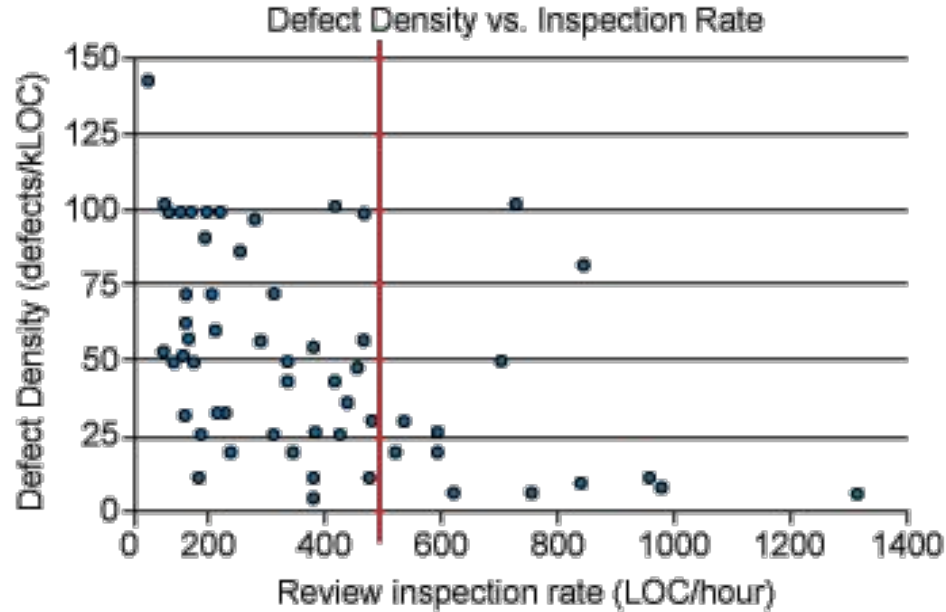
- Increase your bus factor
- Ensure code is readable
- Performance and scalability
- Maintain code standards

# Keep changesets small



Source: Cisco review study, <https://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/>

# Don't review too much too quickly



Source: Cisco review study, <https://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/>


## Ideally




- No silos
- Everyone reviews and all code is reviewed
- Pair programming, collaborative planning




## Practically

- Maximise channels of communication
- Everyone is a potential reviewer
- Be prepared to review unfamiliar code

# Help reviewers know what they're looking for

 **Open** xiangli313 wants to merge 6 commits into `master` from `i1033`

 Conversation **9**  Commits **6**  Files changed **6**

 xiangli313 commented on 23 Nov 2017 • edited by tinigarske ▾  

**Report name**

```
"internal-2017-201710gavi-version2-import"
```

**Link to youtrack issue**

<https://vimc.myjetbrains.com/youtrack/issue/VIMC-1033>

**Instructions for running the report**

```
orderBy::orderBy_run("internal-2017-201710gavi-version2-import")
```

**Things you would particularly like Tini to look out for**

Let me know if there are other things you would particularly like to have for this report.

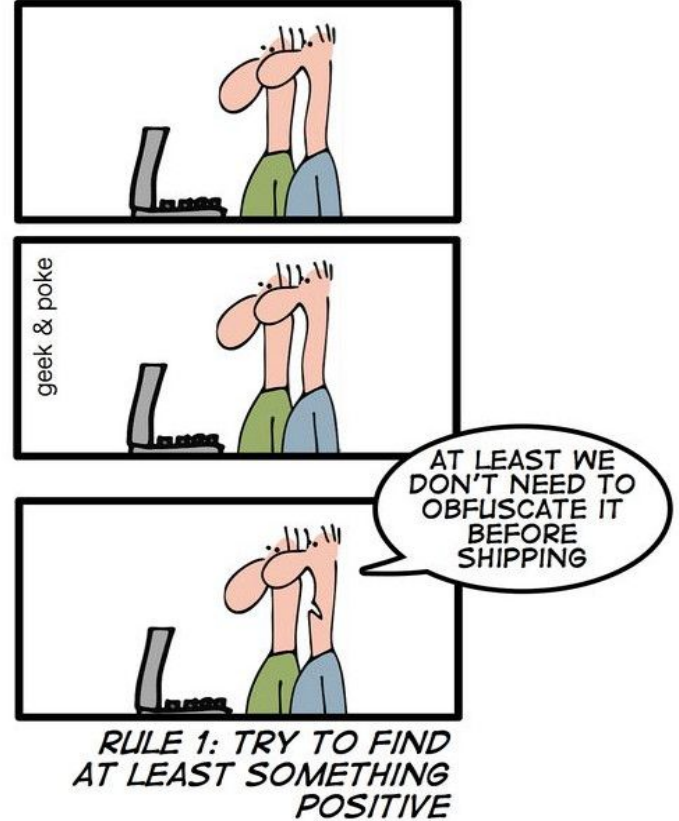
**Things you would particularly like anyone else to look out for**

**Checklist for reviewer**

- is the name sensible?
- is there only one report in here (or does it need more than one report)
- does it run locally on your machine
- does it run on science (`orderBy::orderBy_run_remote("name", ref = "origin/iXXX")`)

# The psychology of code reviews

## HOW TO MAKE A GOOD CODE REVIEW







**Alex**  
@alexhillphd



Hey fellow developers, I'm curious to know, how often do you feel defensive when receiving critical comments on a code review? Please RT! #webdev #softwaredevelopment #agile #codereview #webdevelopment



25 votes • Final results

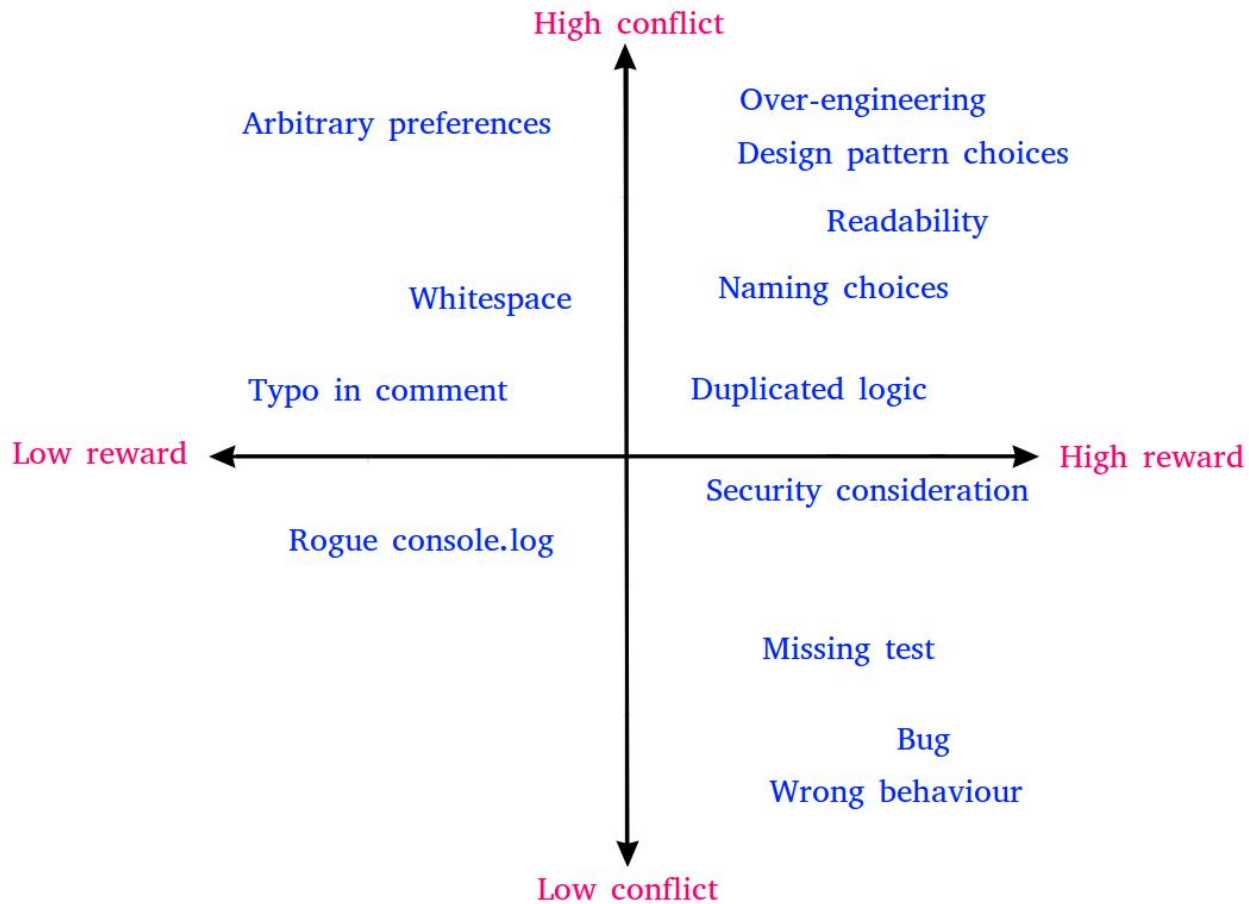
5:41 PM - 27 Feb 2018

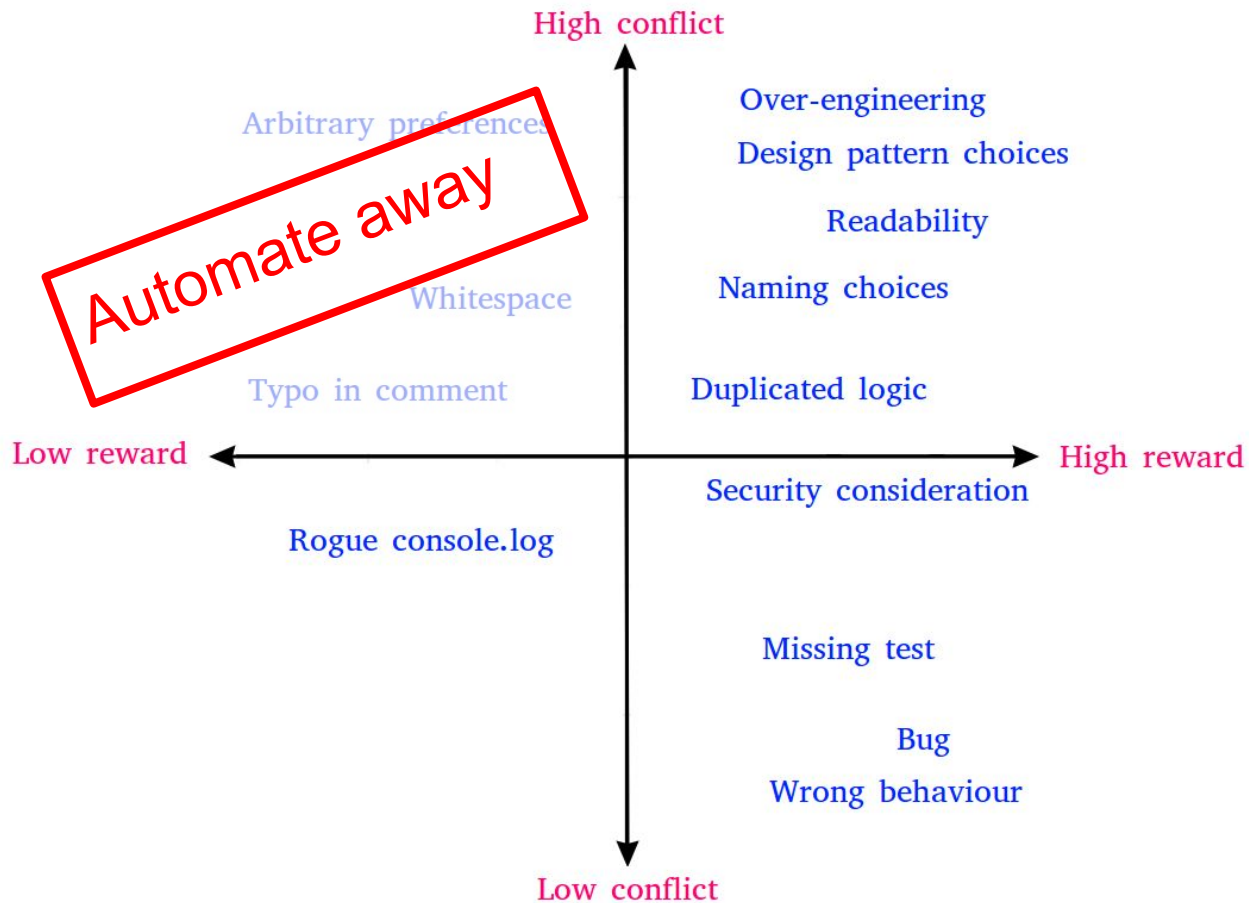
6 Retweets

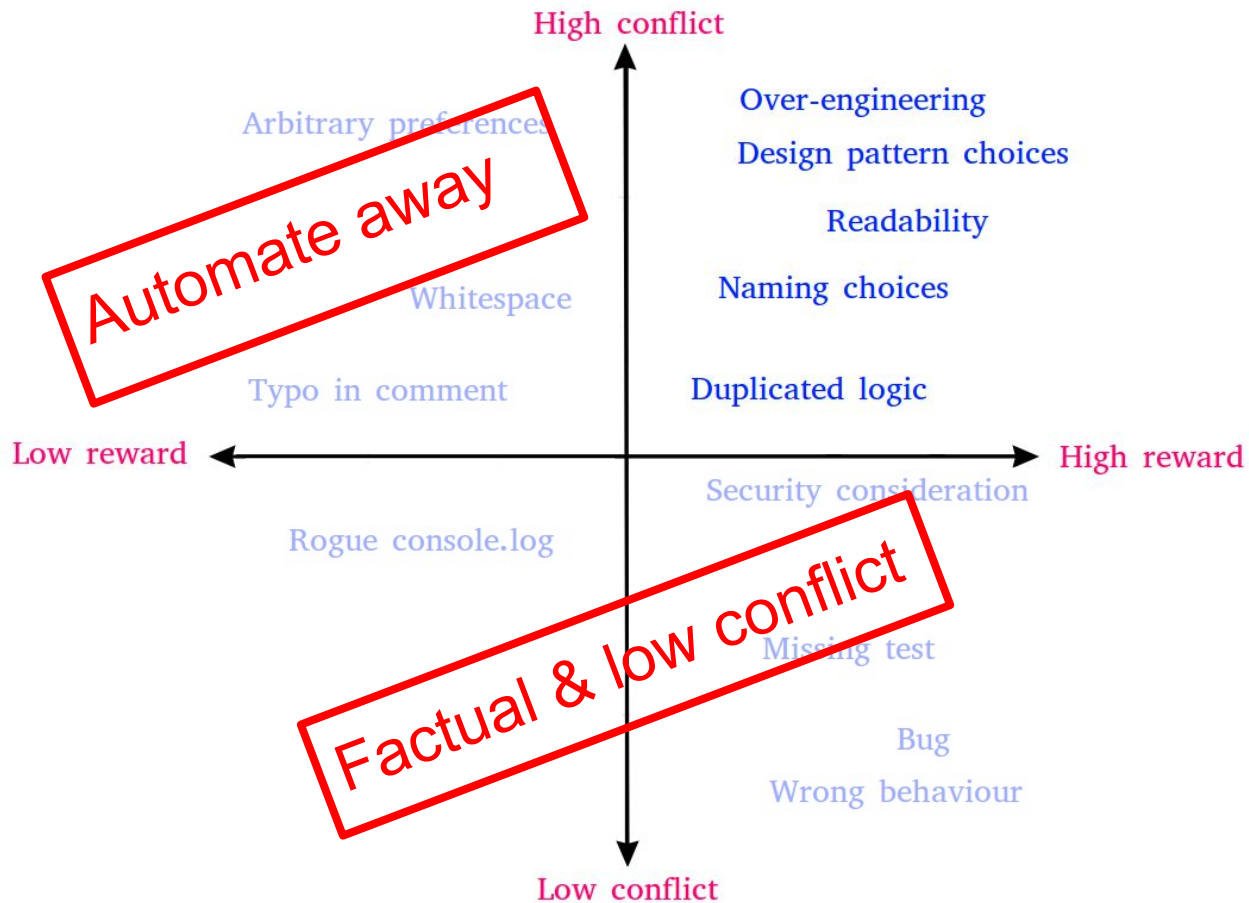


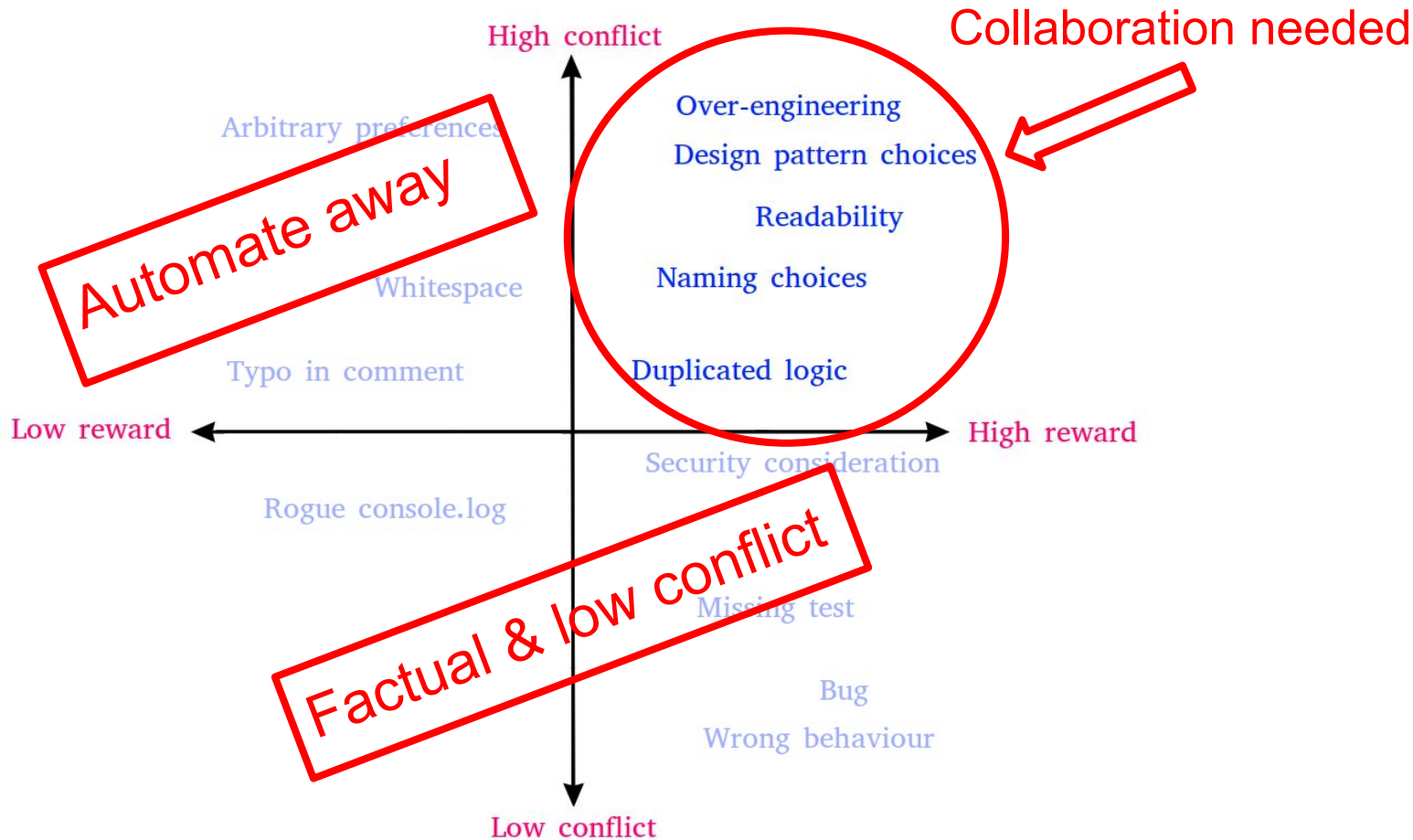
Add another Tweet

expressed too many times that there is always a gold nugget in a neuroimaging

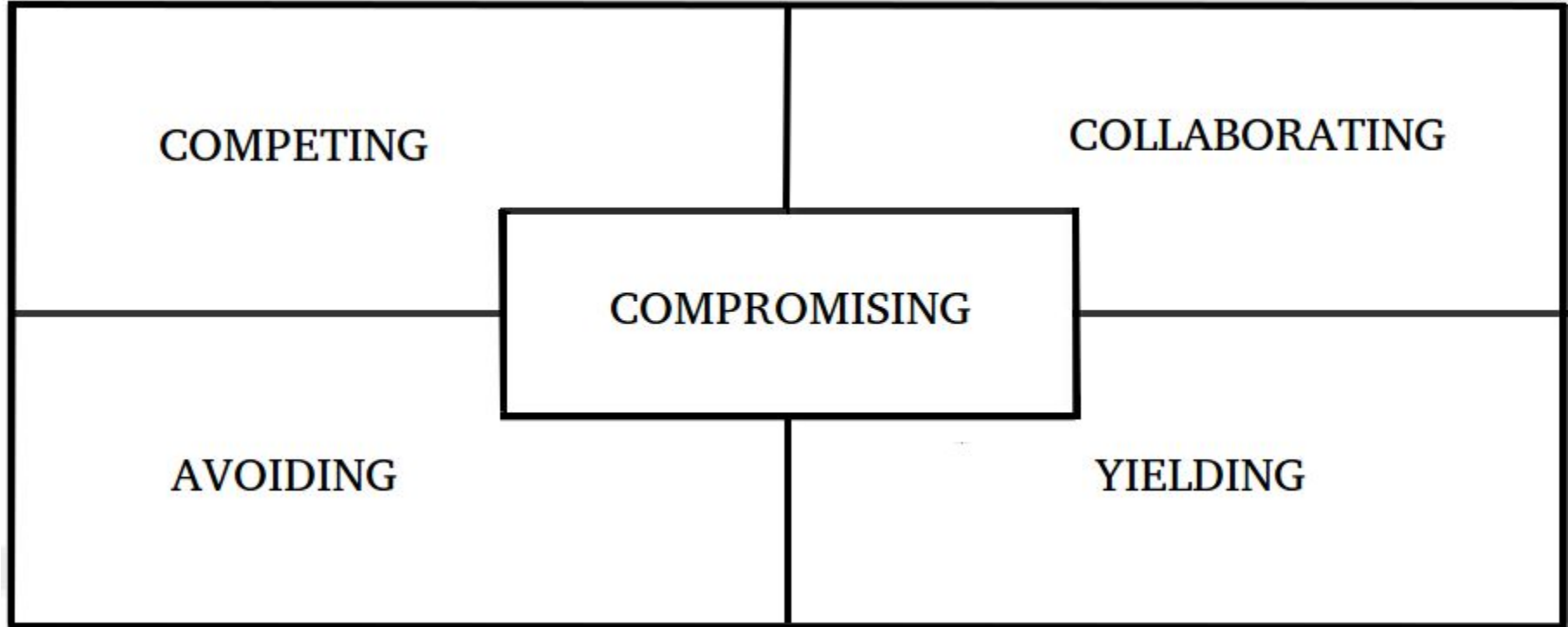






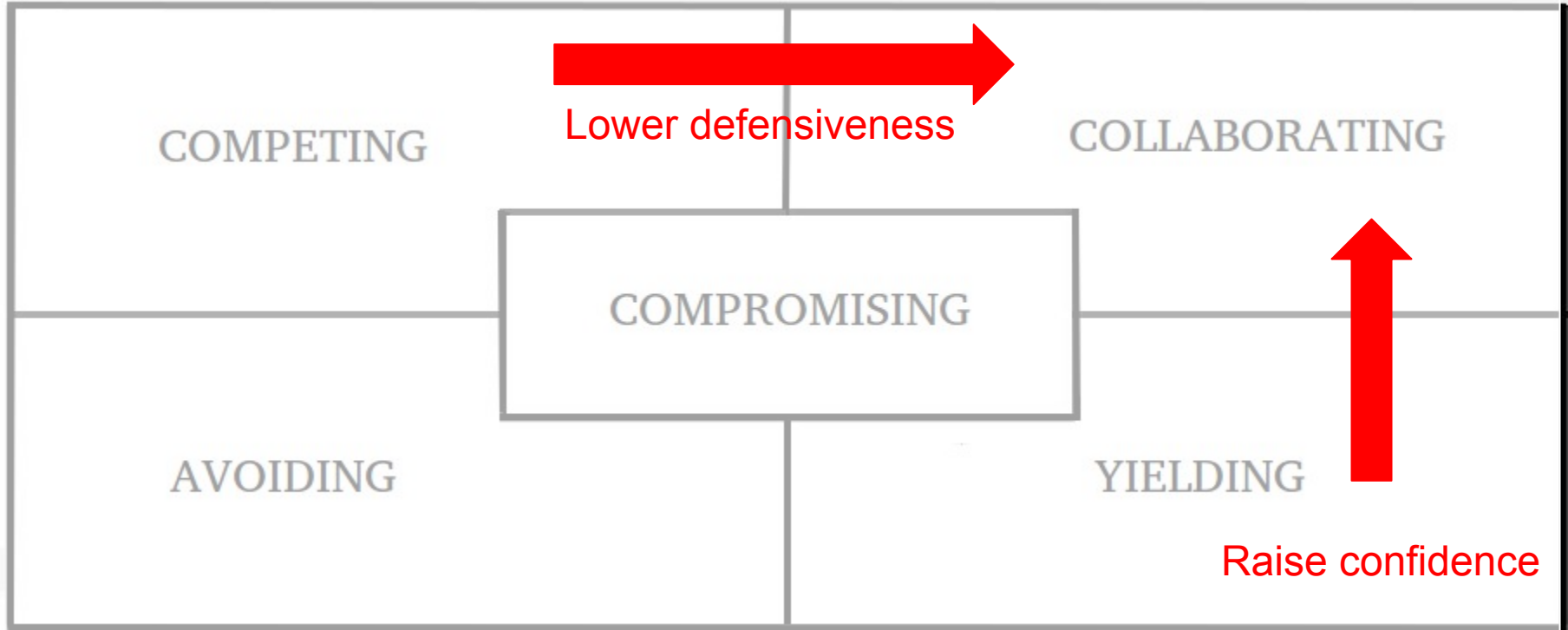


# Conflict resolution archetypes



(Dual concerns model)

# Conflict resolution archetypes



(Dual concerns model)

# As a reviewer we can...

Lower defensiveness ✓

Raise confidence ✓

- Raise code by a grade or 2, no more<sup>1</sup>
  - e.g. C → B+, B → A
- Use 'we' instead of 'you'
  - e.g. "You should have re-used this function" → "We could re-use this function"
  - b) "Why did you do it this way?" -> "What are the advantages of this approach?"
- Ask questions
  - e.g. "We could re-use this function" → "How does this functionality differ from function x?" / "Can we re-use x here?"

1. Credit for this idea from: <https://mtlynch.io/human-code-reviews-2/#aim-to-bring-the-code-up-a-letter-grade-or-two>



# As a reviewer we can...

Lower defensiveness ✓  
Raise confidence ✓

Give positive feedback!

- “This is a cool library you found”
- “This refactor makes a lot of sense”
- “This is really easy to read”
- “I didn’t know this function existed, thanks for bringing it to my attention!”
- “Thanks for your hard work on this”

## As an author we can...

Lower defensiveness ✓

Raise confidence ✓

- Practice the 'as if' technique
  - How would I respond if I were in my best mood?
  - How would I respond if this code was not mine?
  - How would I respond if the comments were phrased differently?
- Say 'thank you'
- Annotate your review first
- Solicit feedback with specific questions

# Thanks for listening!

alex.hill@gmail.com

@alexhillphd