Effective Code Review for Agile Java Developers

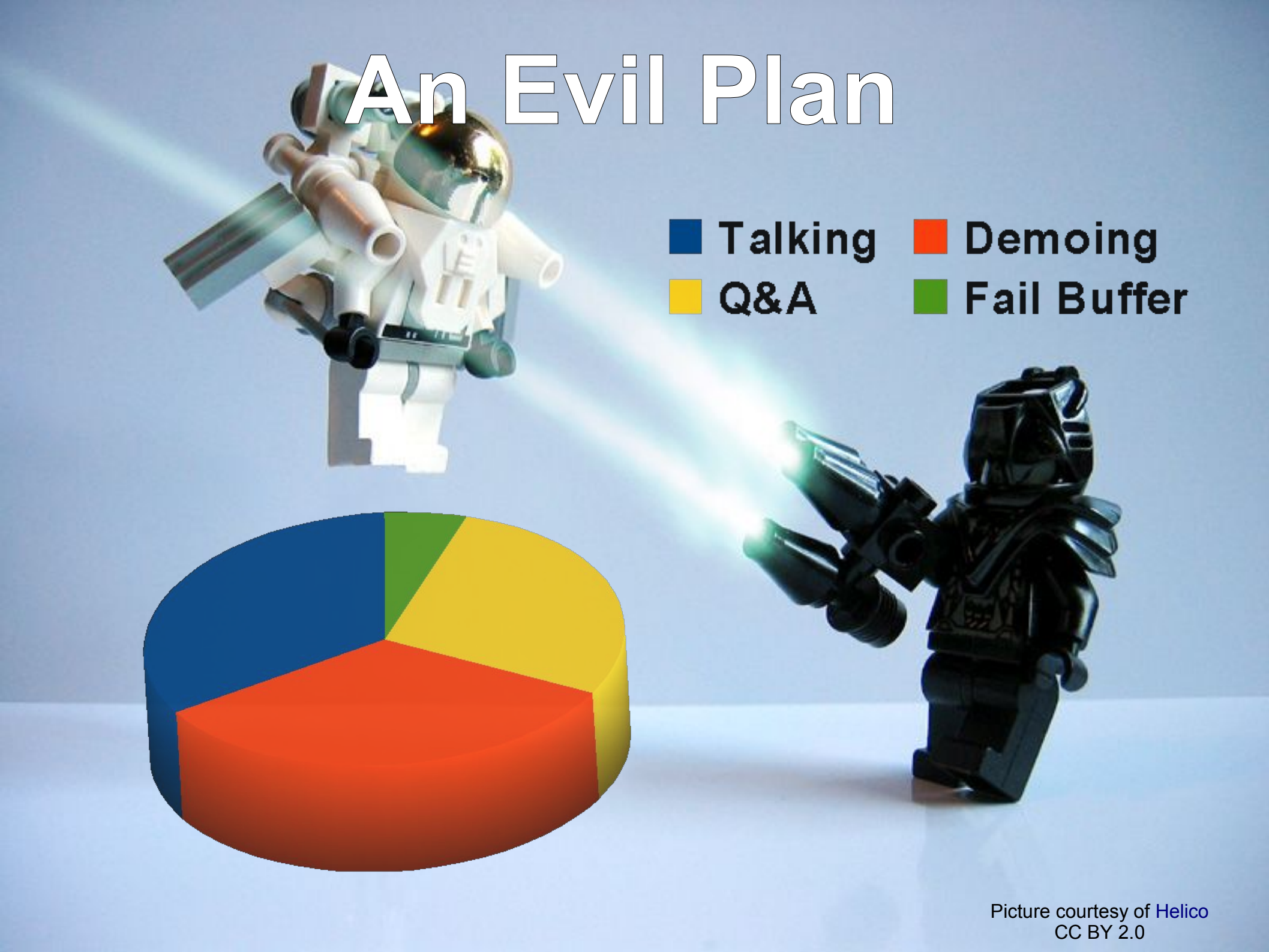
Wojciech Seliga & Sławomir Ginter

JDD Java Developers' Day

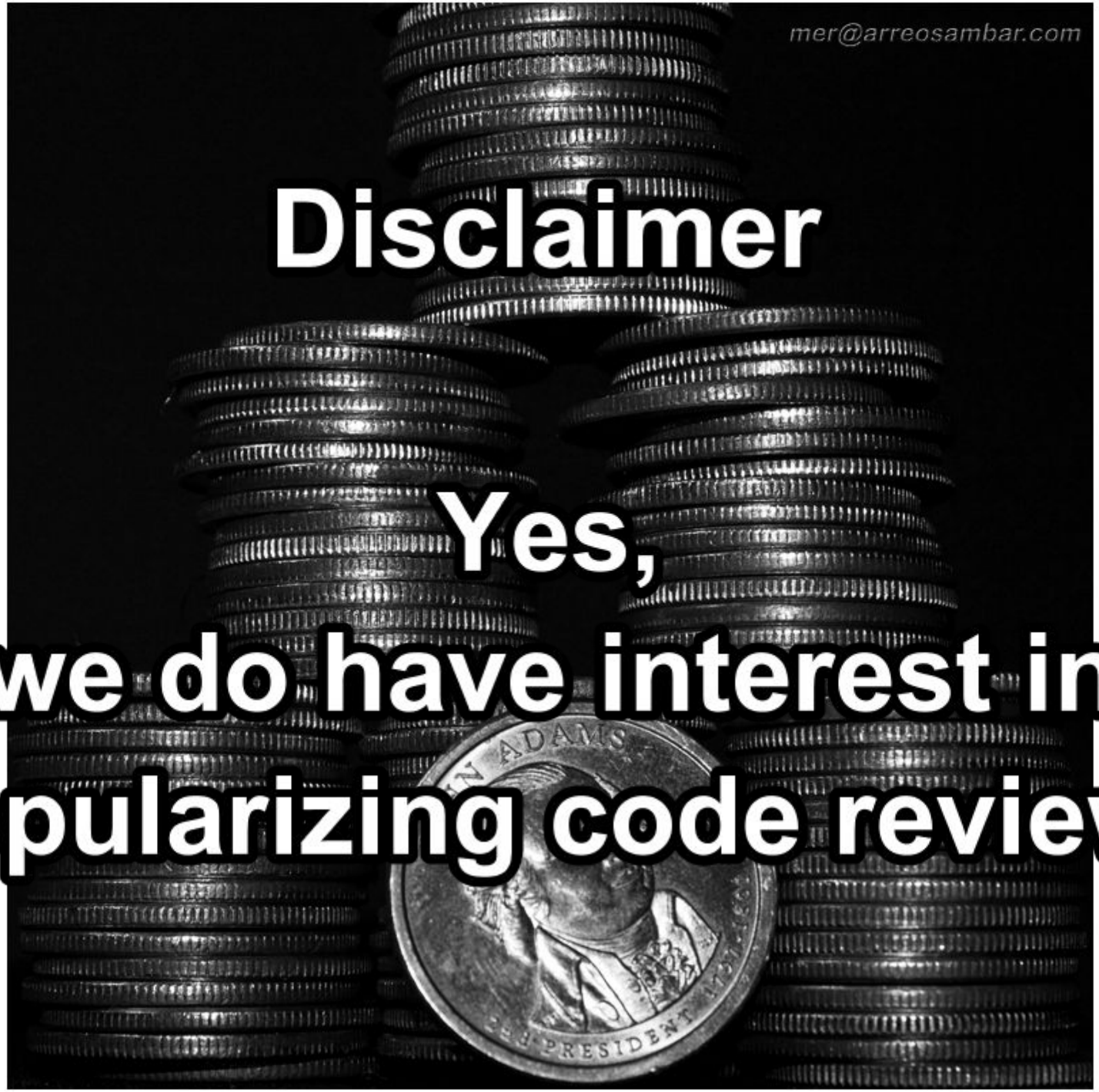
Licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License.

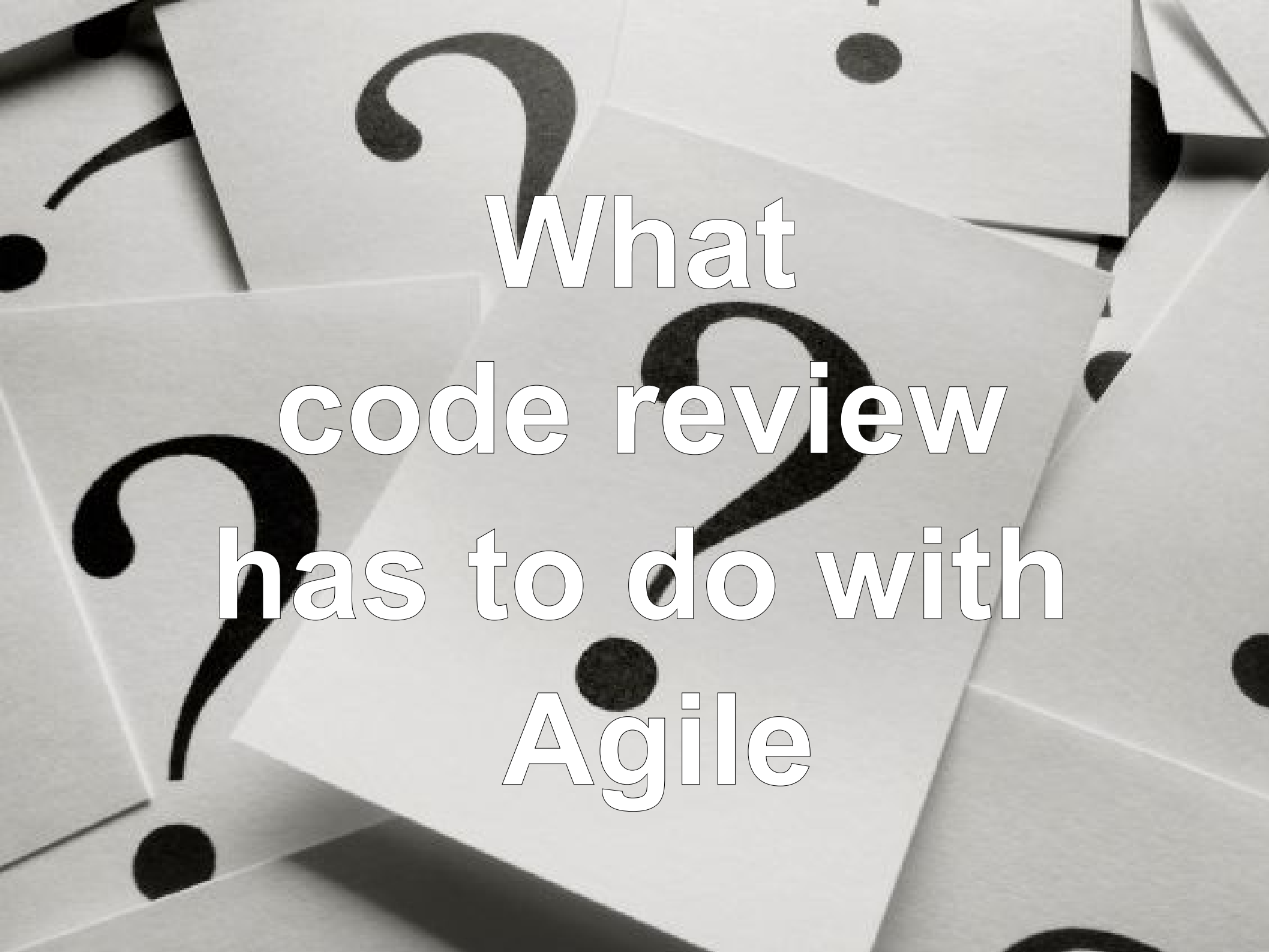
Tag cloud generated with http://www.wordle.net/

An Evil Plan

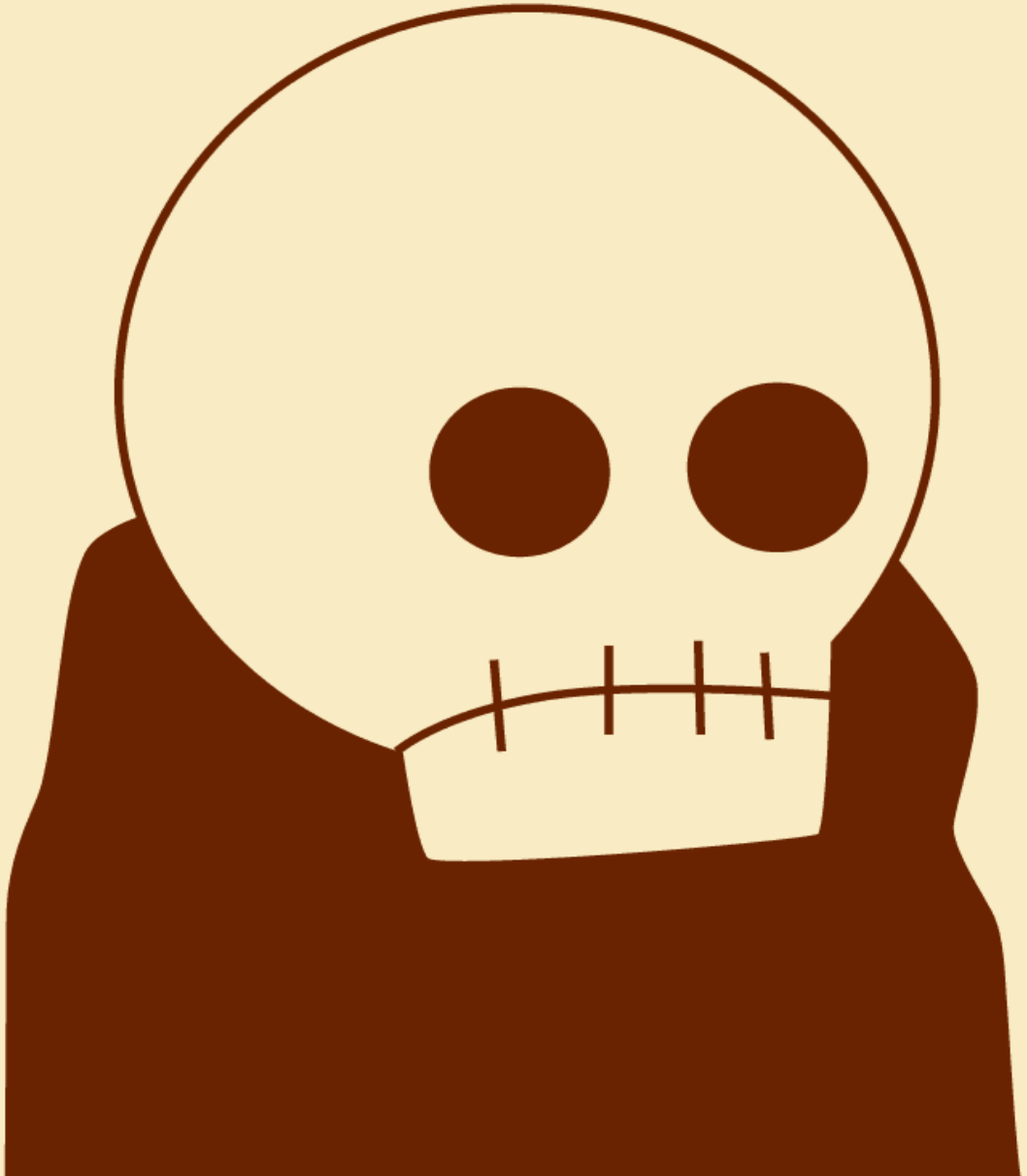


Picture courtesy of Helico
CC BY 2.0


mer@arreosambar.com
Disclaimer
Yes,
we do have interest in
popularizing code reviews


What
code review
has to do with
Agile

WHY BOTHER



Picture courtesy of Cat and Girl
CC A-NC-SA 2.5

Picture courtesy of mil8 / CC BY 2.0

Mentoring junior developers

Non-intrusive

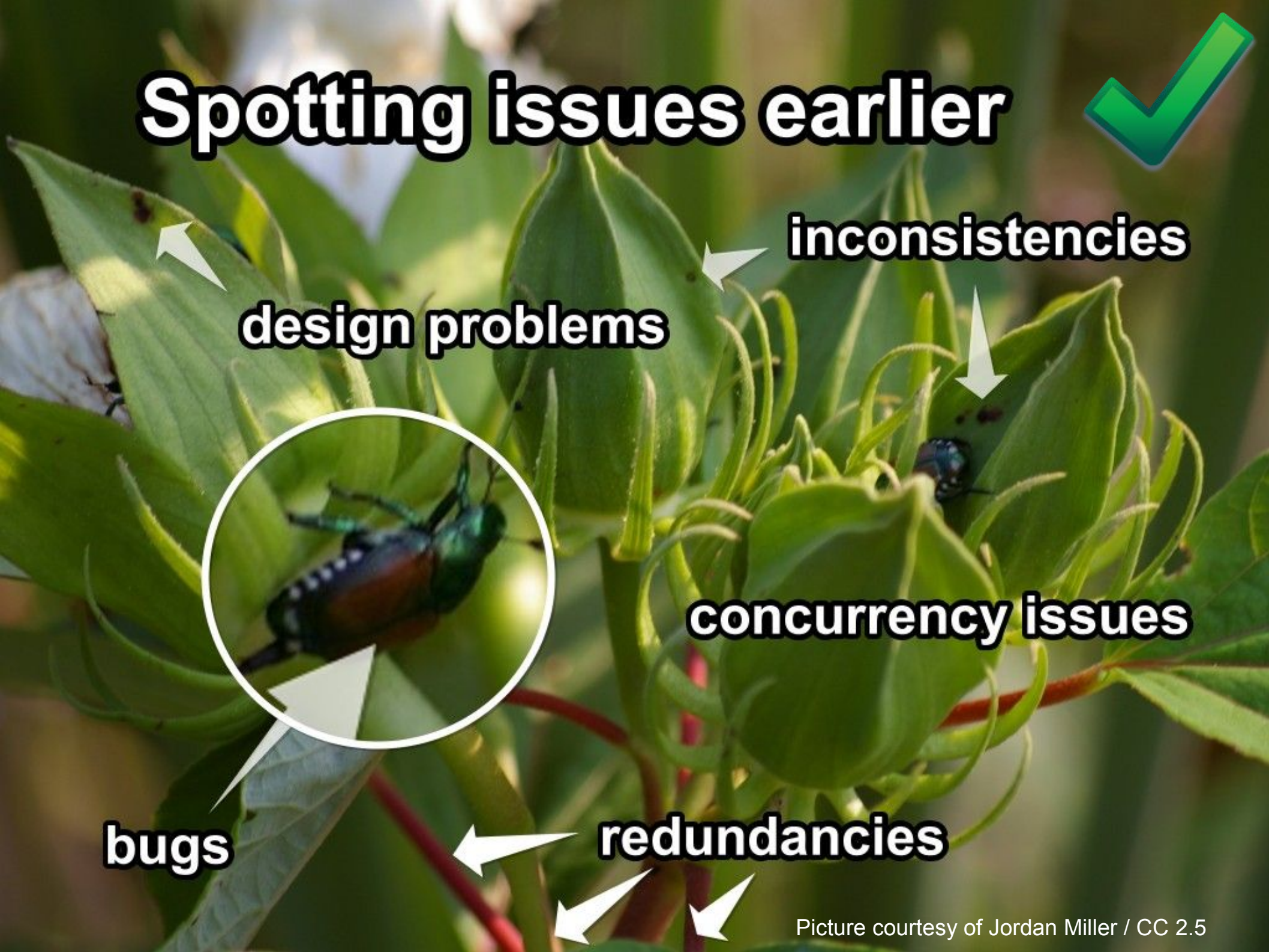
Asynchronous

Less frustration / interruption for senior devs

Picture courtesy of eddidit / CC BY 2.0

Sharing good engineering practices

Picture courtesy of Hey Paul / CC BY 2.0



Picture courtesy of Jordan Miller / CC 2.5

Facilitating distributed teams

Photo Courtesy of U.S. Army


Why then is code review so rarely used?
Picture courtesy of tinyfroglet / CC 3.0

Preparation is difficult

selecting the code

organizing reviewers

booking conference room

scheduling

printing

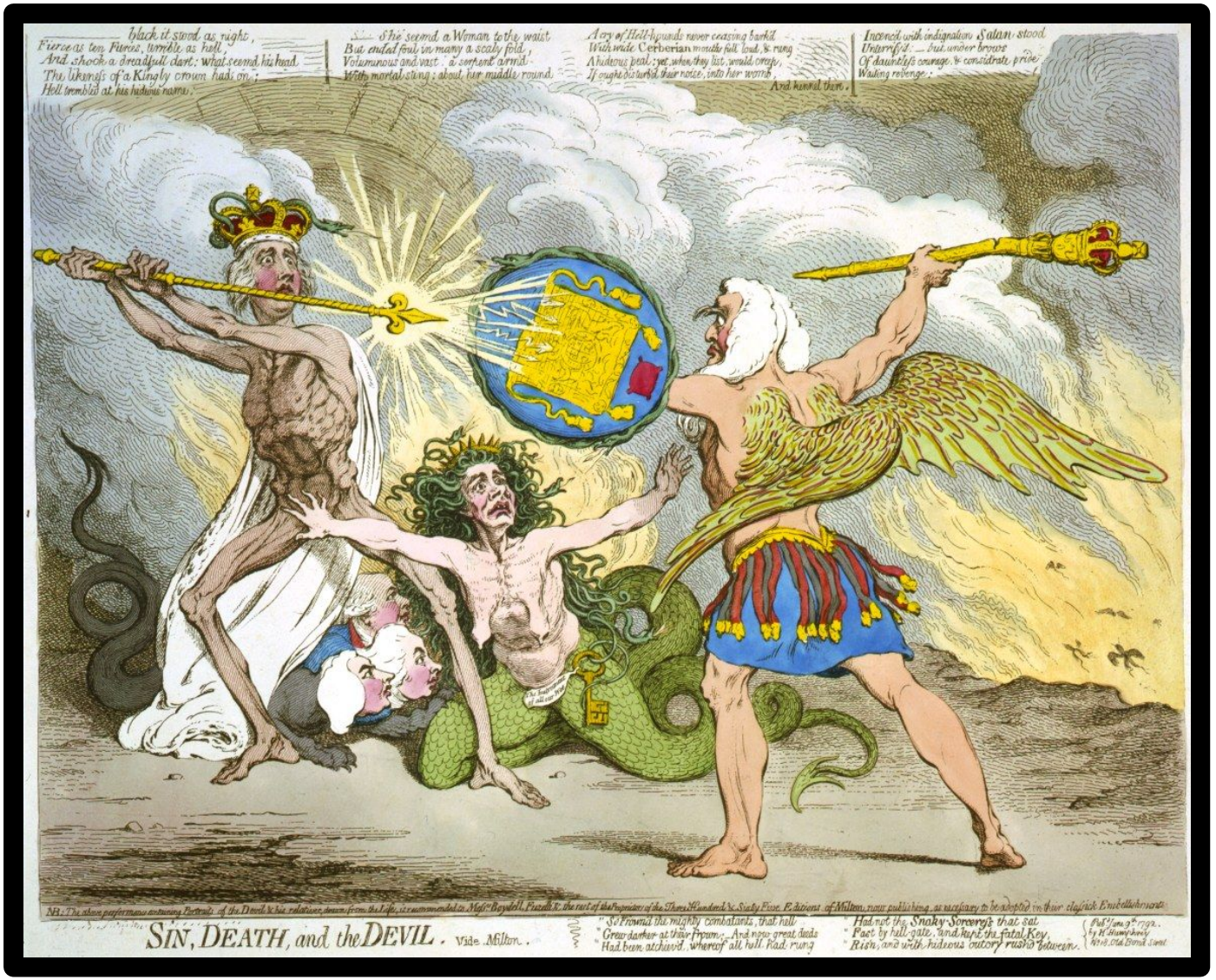
...

Picture courtesy of jfdervin / CC BY-SA 2.0


Yet another disruption ?
synchronized meetings
de-focusing

Picture courtesy of gadl / CC BY-SA 2.0

Risk of animosities



No concrete measureable results

Picture courtesy of aussiegall / CC BY 2.0

Successful code review

Lightweight - simple & flexible process

Asynchronous

Continuous

Efficient tool support

Diff-oriented whenever applicable

Transparent and persistent

Time for a longer demo

+

Agile code review misconceptions

a fanatic bug-hunt

false confidence about no bugs left

tracking results of every single comment

expecting hard metrics

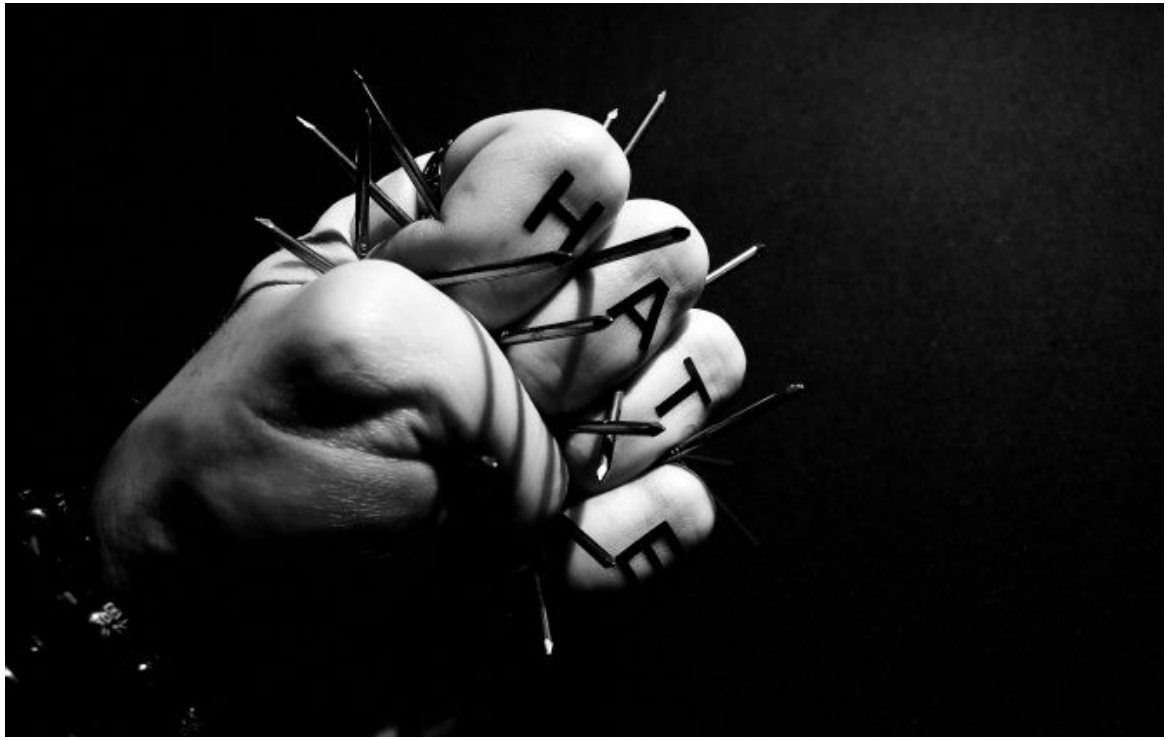
Picture courtesy of Juria Yoshikawa / CC BY-SA 2.0

The greatest initial Enemies

Rigid Process

Metrics

Micro-management



But teams evolve...

Picture courtesy of tms. / CC BY-NC-ND 2.0

Some rules of agile code review

everyone can join review & comment

everyone can modify the scope of the review

everyone can invite other people

everything is public across the company

it's about learning, it's not about blaming

Picture courtesy of PantoDX / CC BY 2.0

Unexpected advantages

Facilitation of distributed teams

Collaboration on low level design

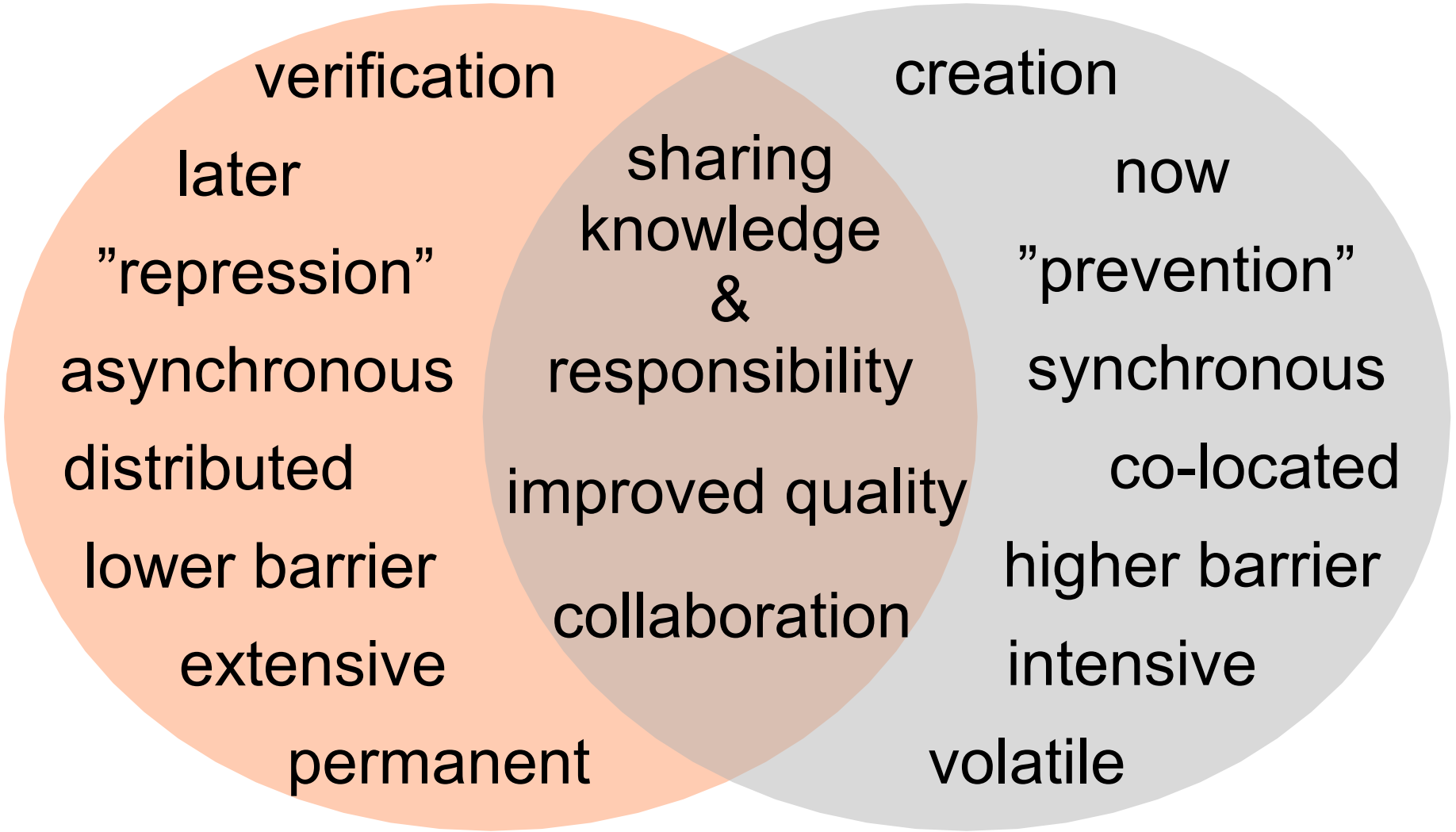
Easier to introduce than pair-programming

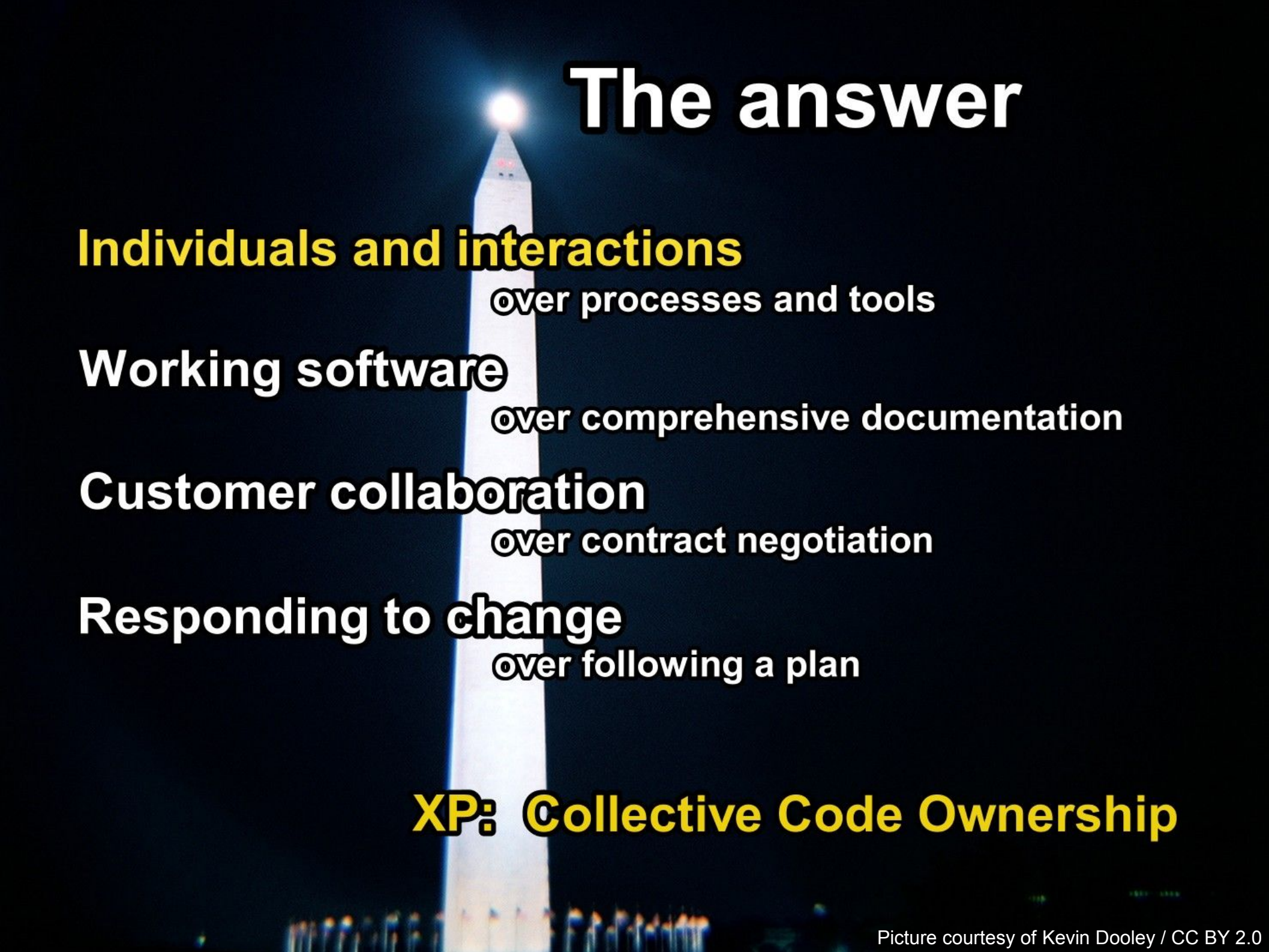
Time-zone difference may help you

Knowledge Base

Picture courtesy of david.nathan.cox / CC BY 2.0

Code Review vs. Pair Programming





The answer

Individuals and interactions
over processes and tools

Working software
over comprehensive documentation

Customer collaboration
over contract negotiation

Responding to change
over following a plan

XP: Collective Code Ownership

Picture courtesy of Kevin Dooley / CC BY 2.0

About us

wojciech.seliga@spartez.com

slawomir.ginter@spartez.com

Tag clouds generated with http://www.wordle.net/

Q&A

Picture courtesy of Mykl Roventine / CC BY 2.0

Thank you