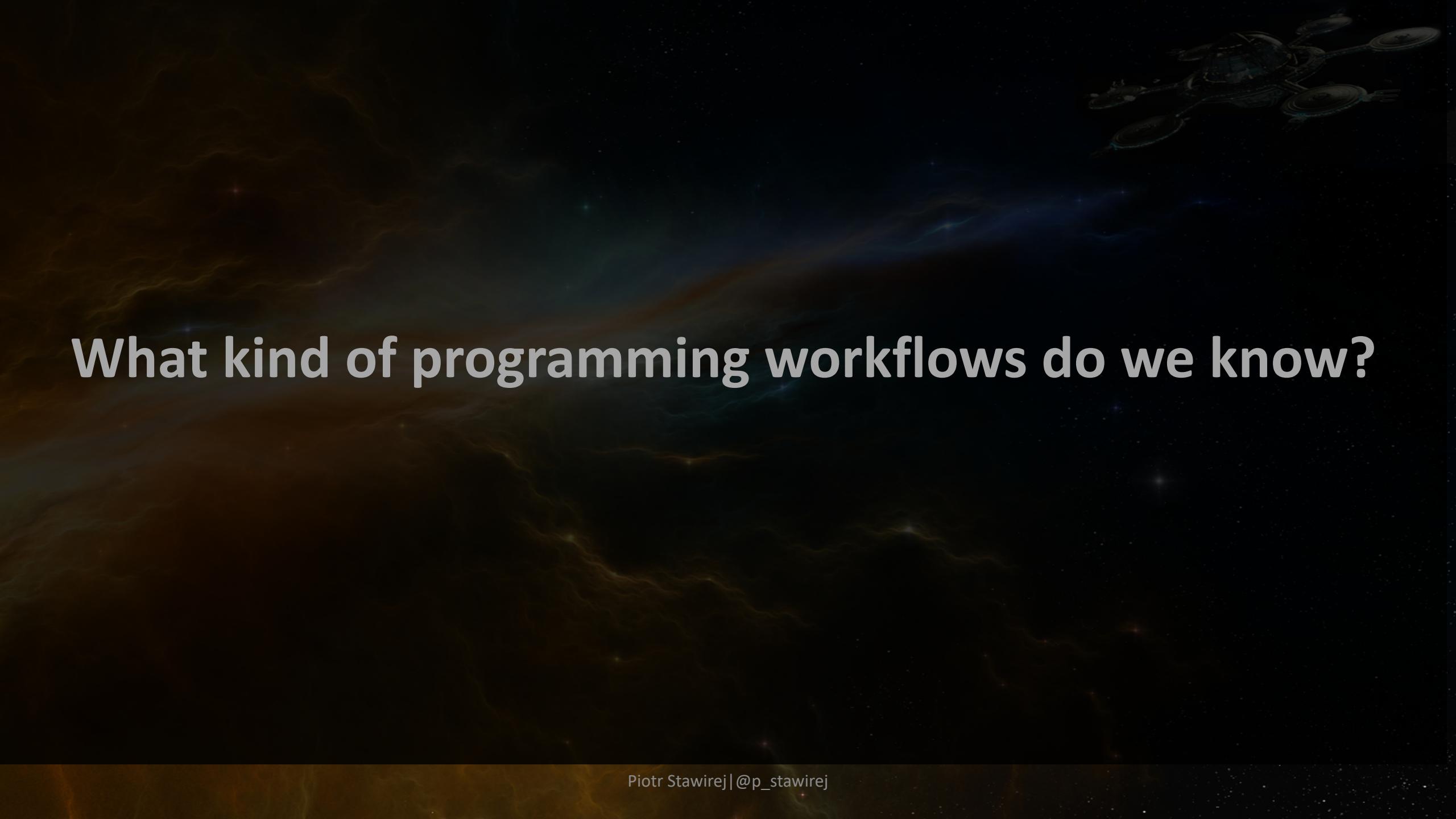


test && commit || revert

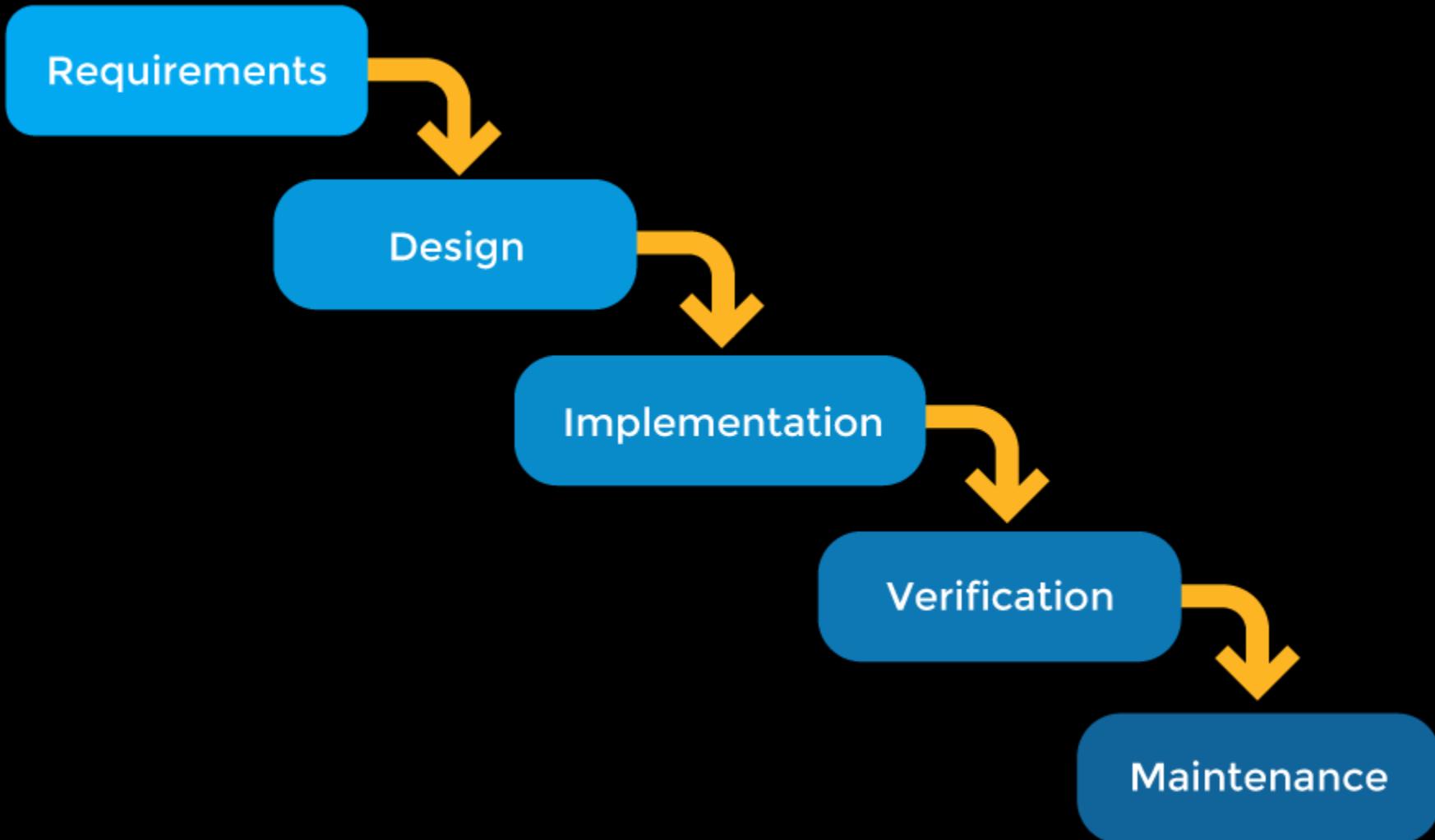
(TCR)

alternative to TDD or eccentricity?

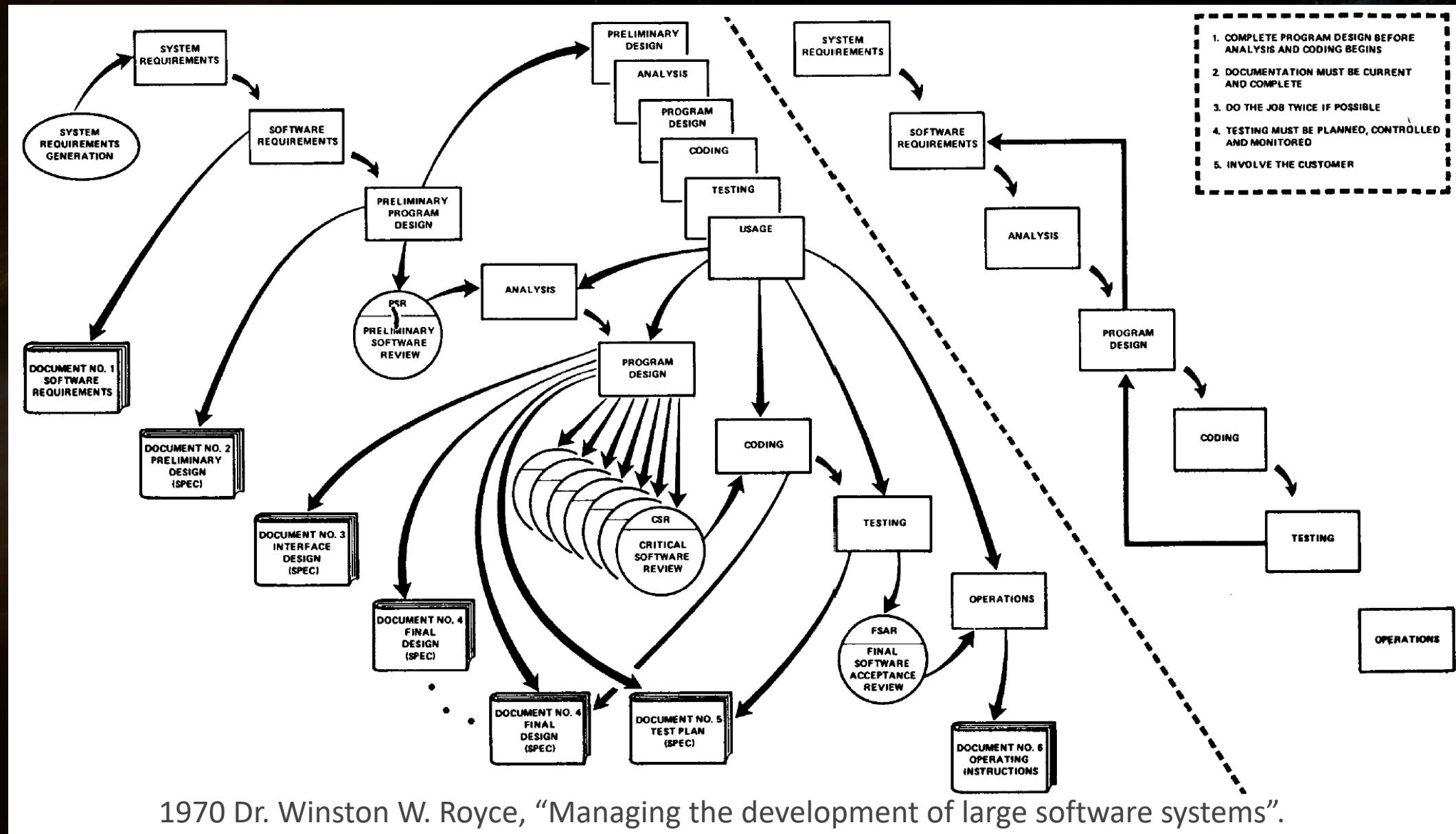


What kind of programming workflows do we know?

Waterfall



Waterfall



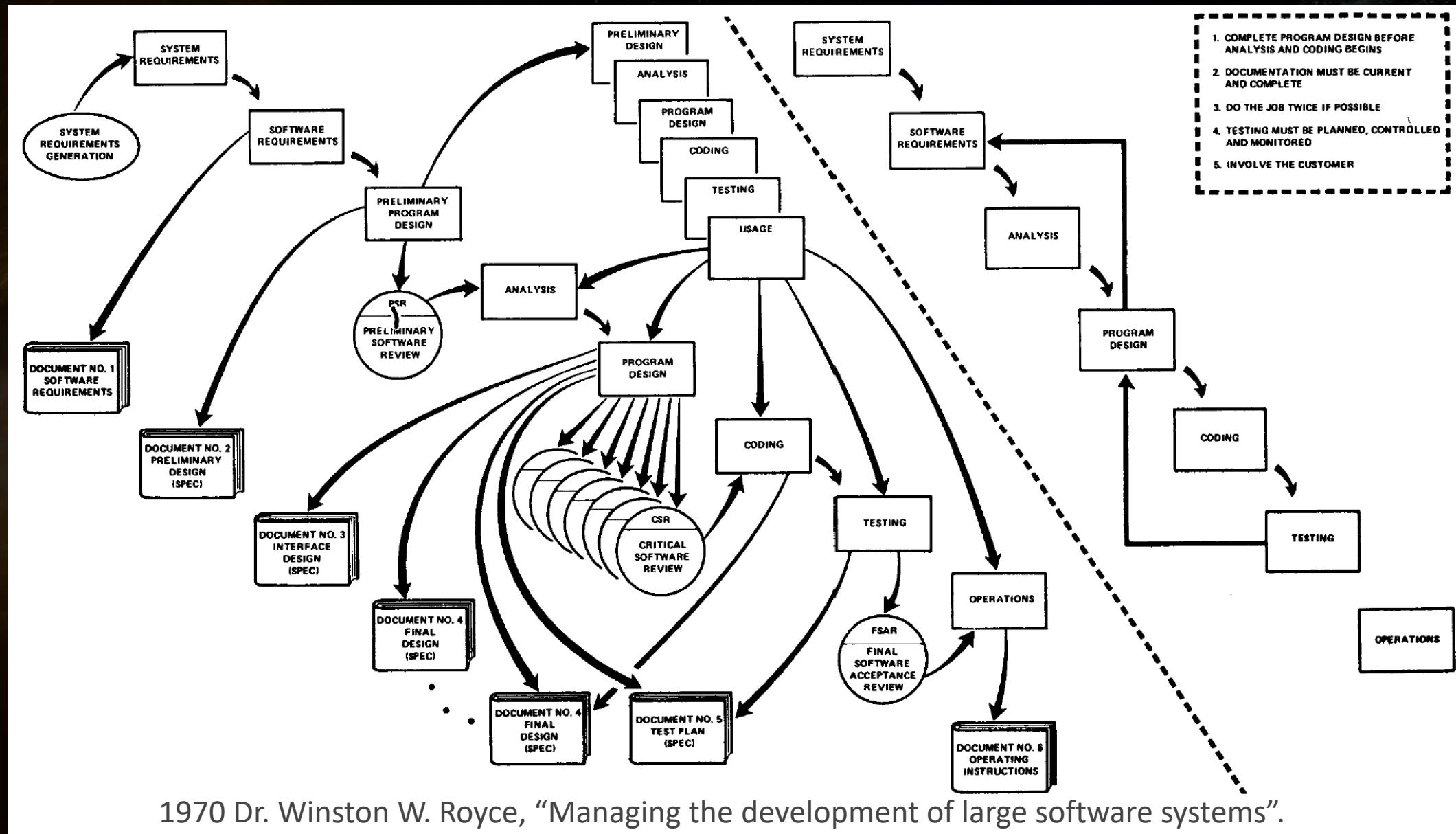
1970 Dr. Winston W. Royce, "Managing the development of large software systems".

Waterfall

- 1. COMPLETE PROGRAM DESIGN BEFORE ANALYSIS AND CODING BEGINS
- 2. DOCUMENTATION MUST BE CURRENT AND COMPLETE
- 3. DO THE JOB TWICE IF POSSIBLE
- 4. TESTING MUST BE PLANNED, CONTROLLED AND MONITORED
- 5. INVOLVE THE CUSTOMER

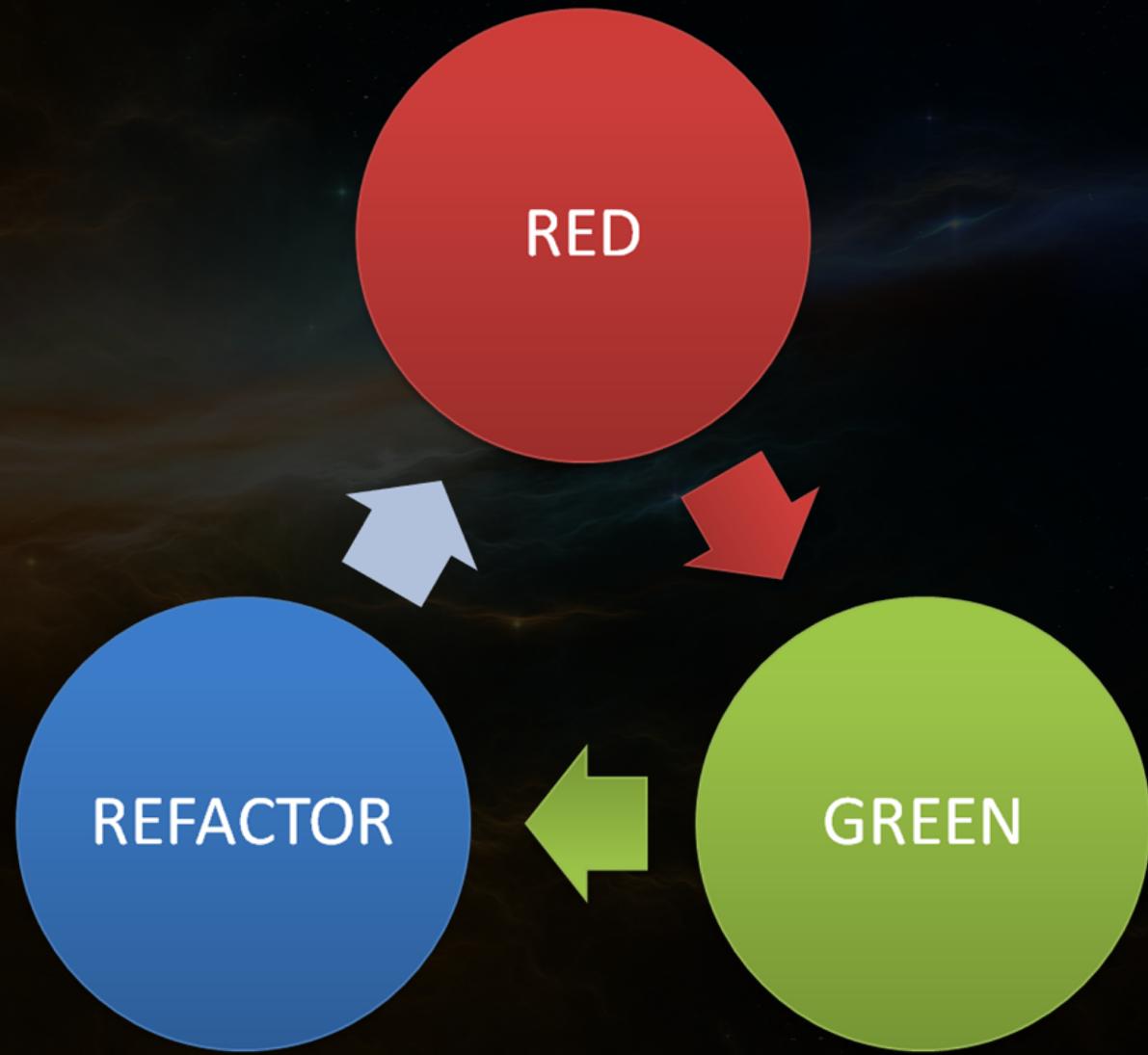
1970 Dr. Winston W. Royce, "Managing the development of large software systems".

Waterfall



1970 Dr. Winston W. Royce, "Managing the development of large software systems".

TDD/BDD





Test after code

Protoduction



Piotr Stawirej | @p_stawirej

Nightmare

Make a change
It doesn't work
Make a change
It doesn't work
Make a change
It doesn't work



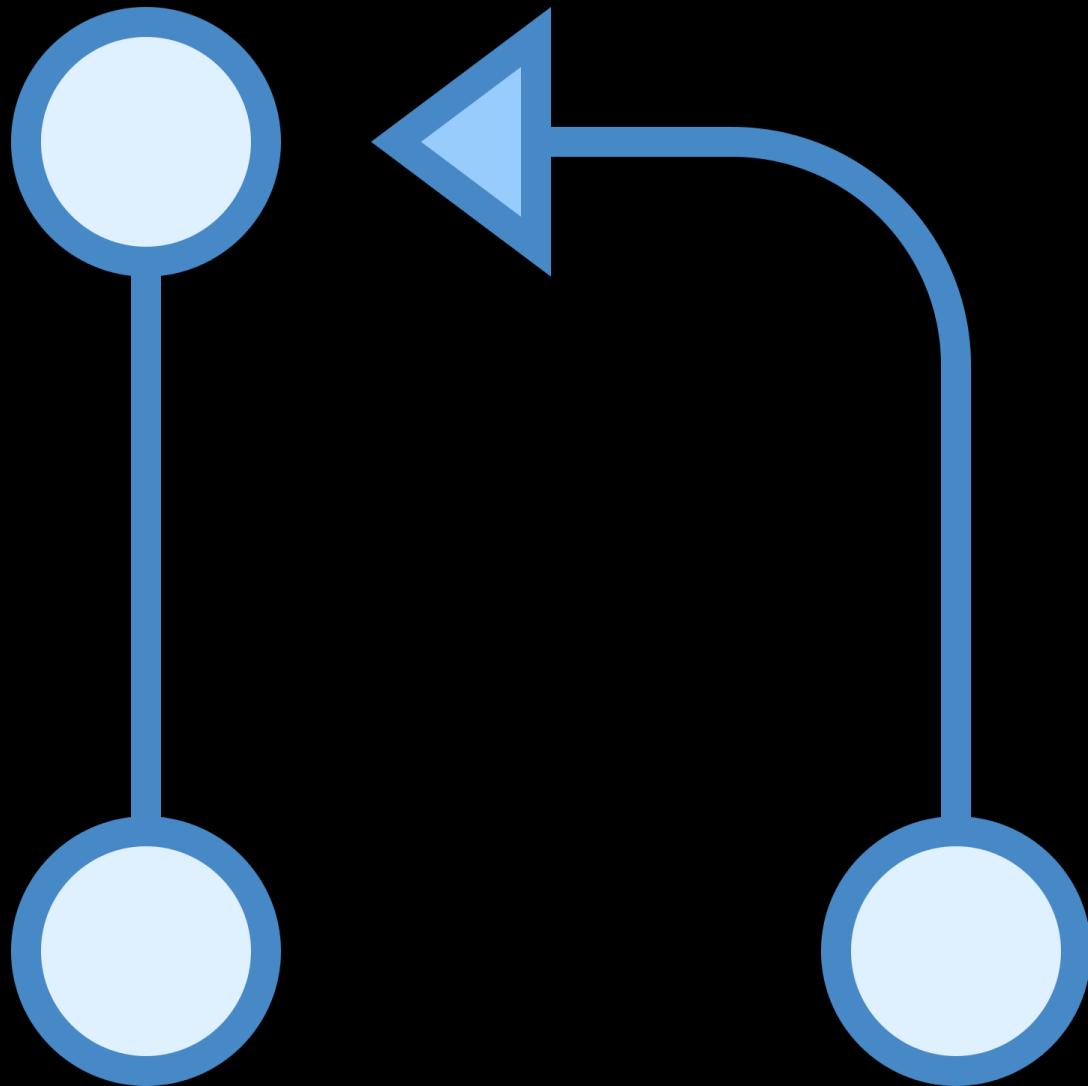


**NO
FEAR**

Any problem - revert to last known green state

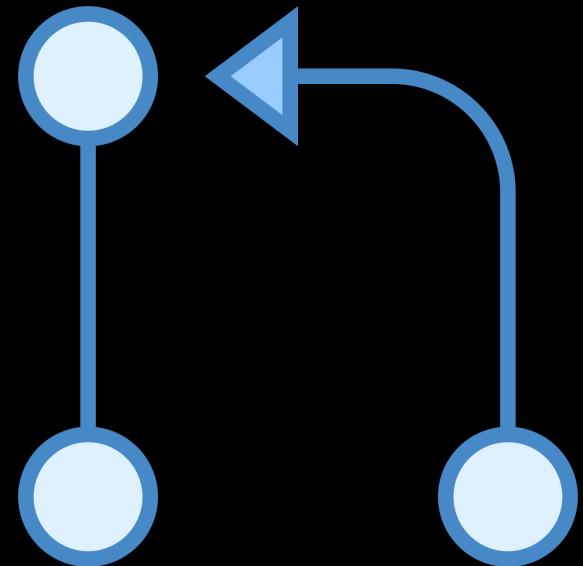
Cheap to get to known state

Big Pull Request

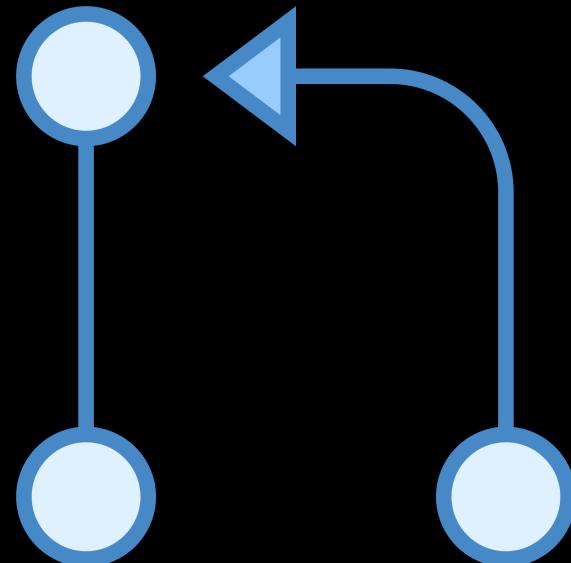


Big Pull Request

What does it mean small PR?



Big Pull Request

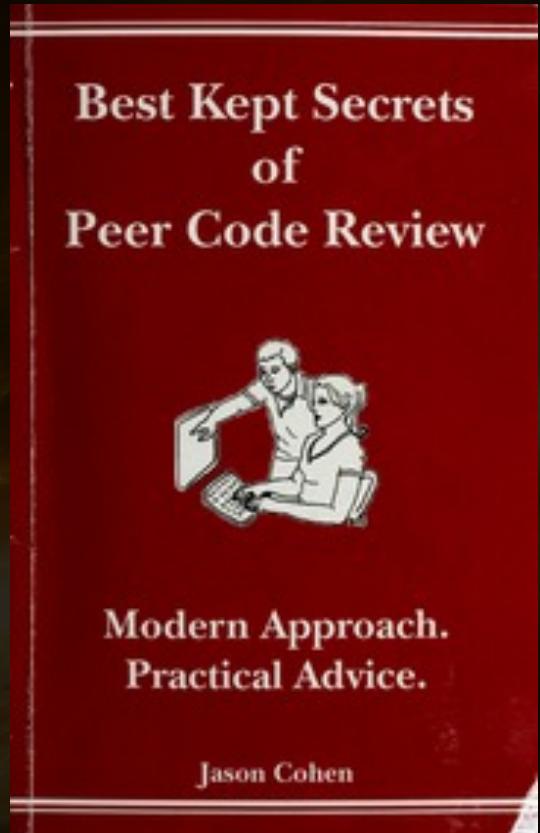


What does it mean small PR?

~400 LOC/h

Brain saturation after 1 hour

200-400 LOC



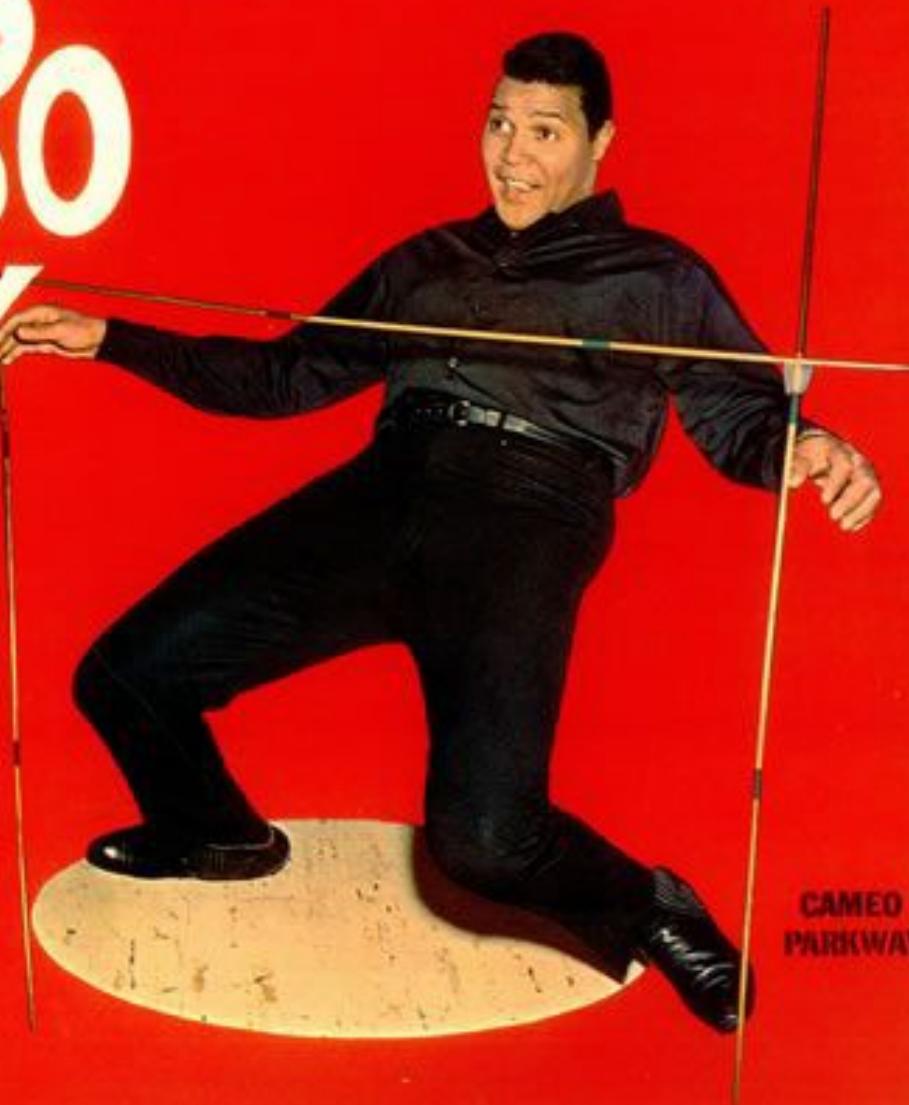
CHUBBY CHECKER

LIMBO

PARTY

THE DANCE CRAZE
THAT'S SWEEPING
THE COUNTRY!

NOW...
HAVE YOUR OWN
LIMBO PARTY!



CAMEO
PARKWAY



test && commit || revert

is a programming workflow where all code that passes your tests gets committed, but if the tests fail all the code written since the last commit will be gone.



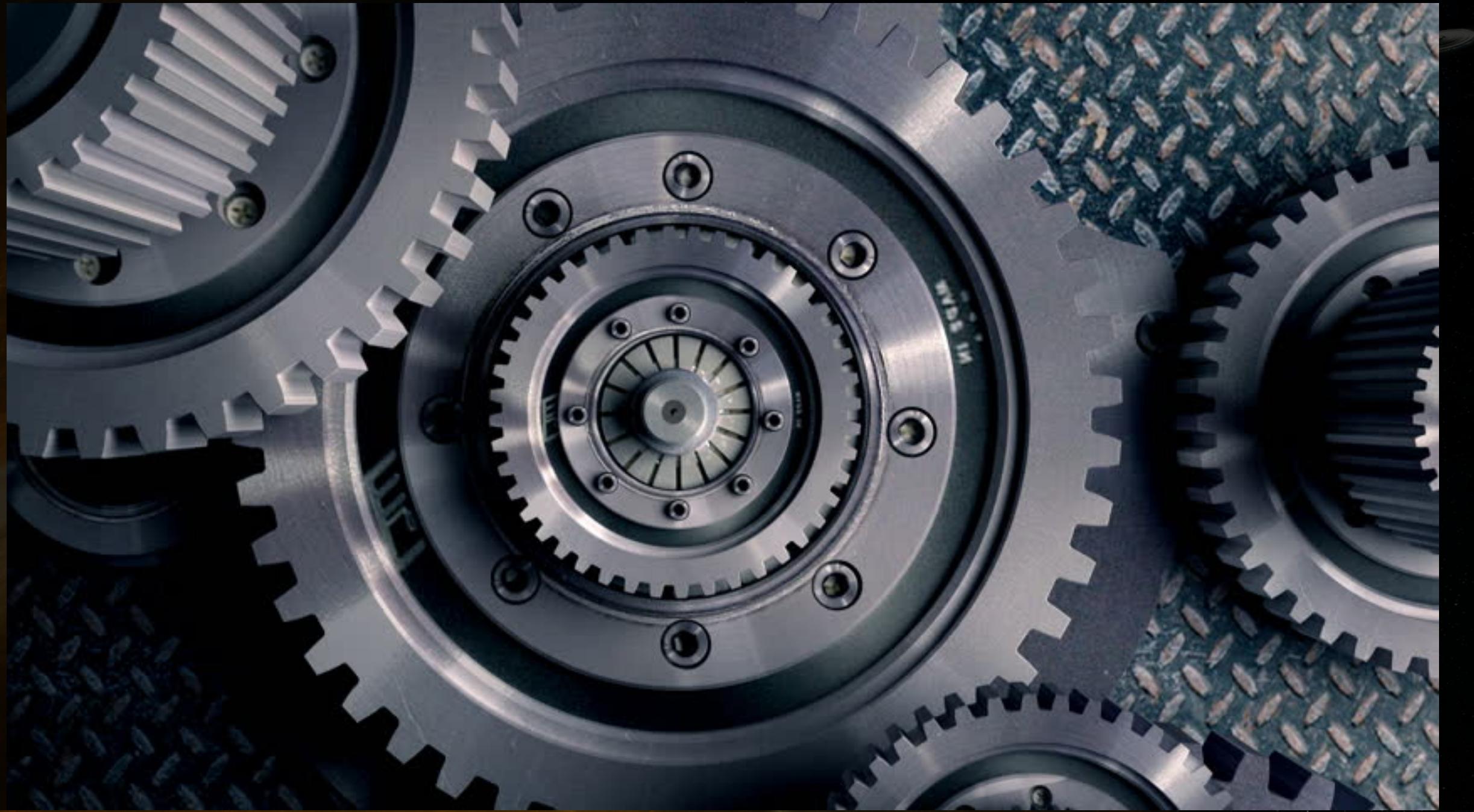
test && commit || revert
is a programming workflow where **all**
code that passes your tests gets
committed, but if the tests fail all the
code written since the last commit will
be gone.



test && commit || revert
is a programming workflow where all
code that passes your tests gets
committed, but if the tests fail all the
code written since the last commit will
be gone.

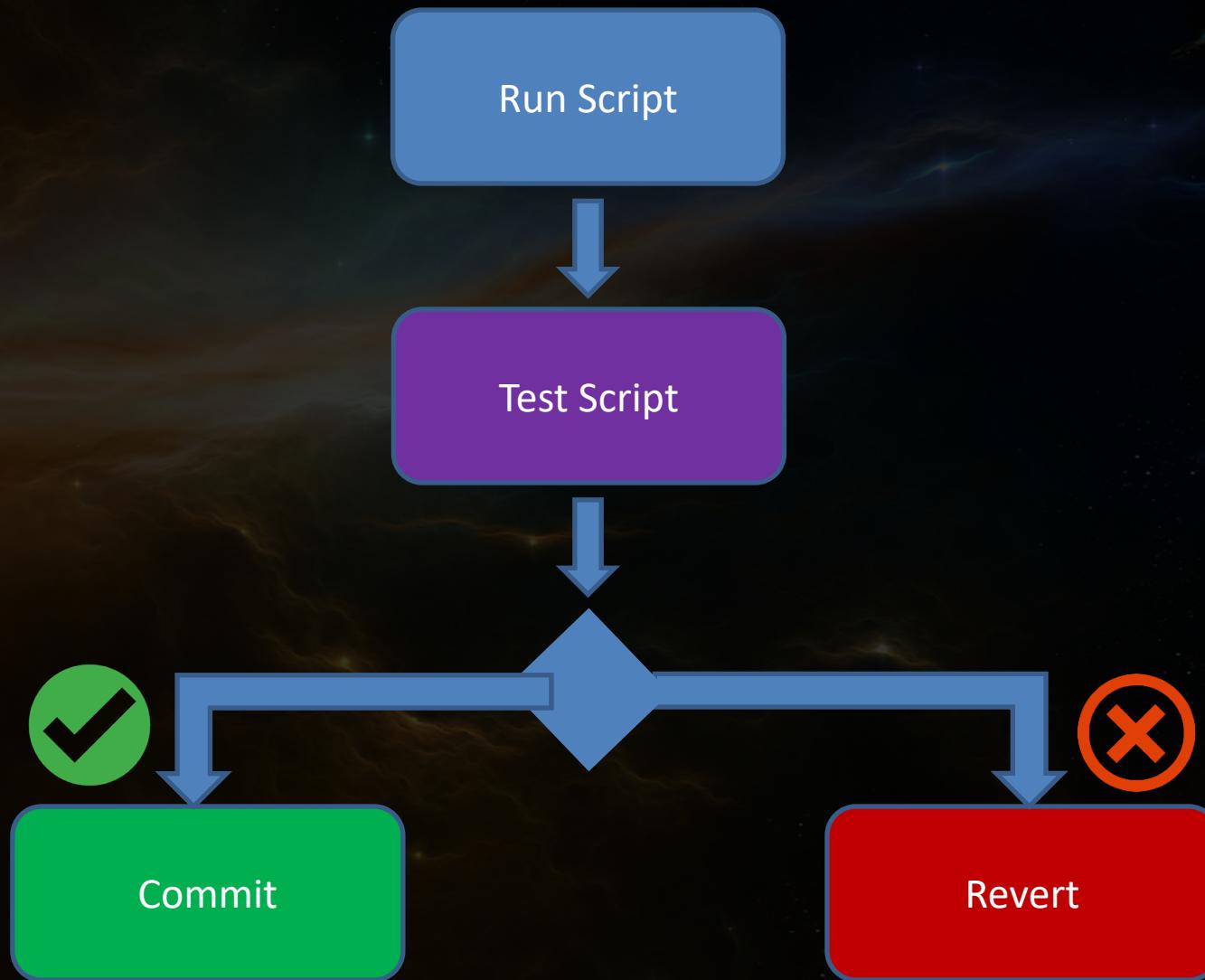


DEMO

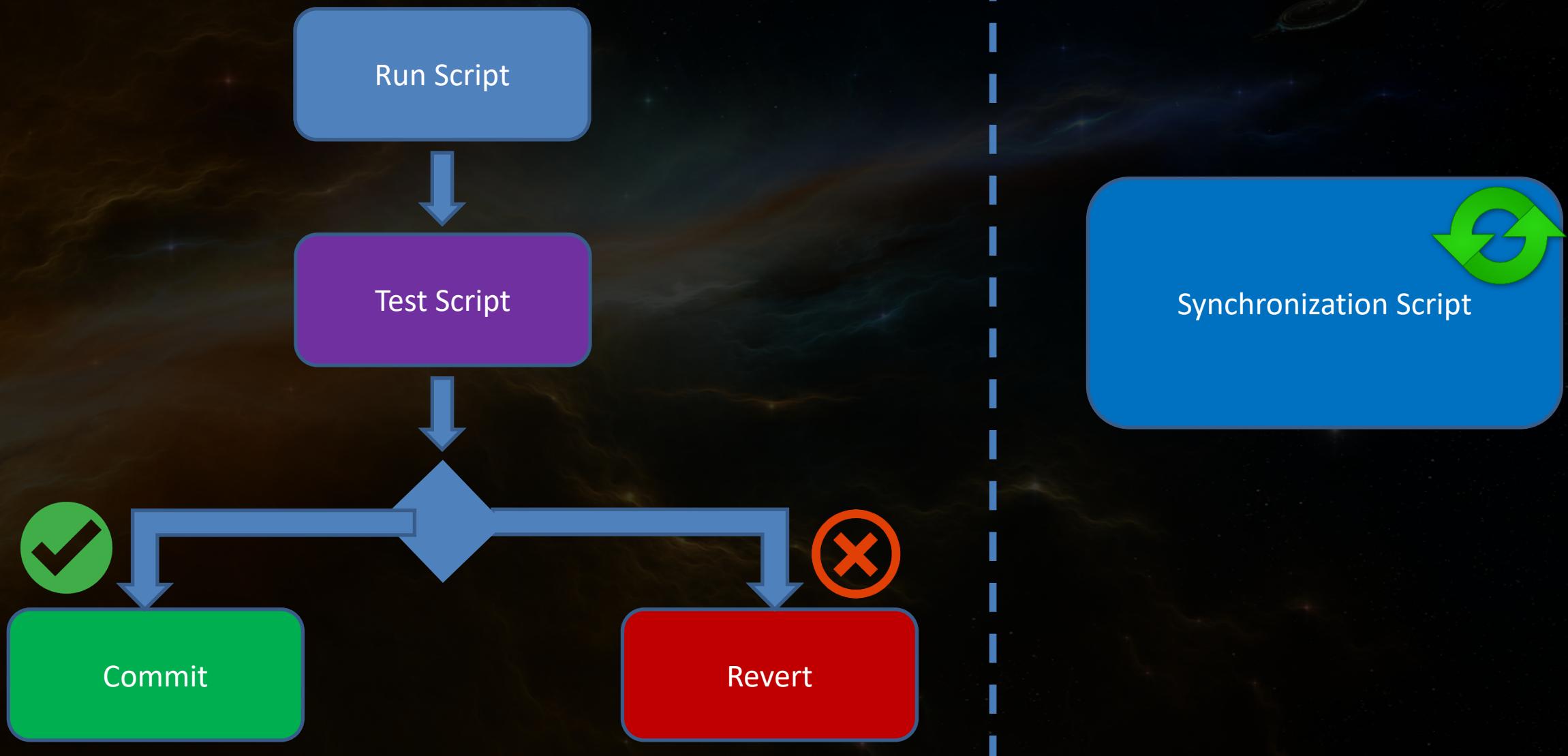


Piotr Stawirej | @p_stawirej

Run script



Run script



Synchronization script



```
#!/usr/bin/env bash

while(true); do
    git pull --rebase;
    git push;
done;
```

Test script



```
#!/usr/bin/env bash
```

```
mvn test
```

Run script



```
./test && git commit -am working || git reset --hard
```

Run script revisited



```
#!/usr/bin/env bash

test() {

}

commit() {

}

revert() {

}

test && commit || revert
```

Run script revisited

```
#!/usr/bin/env bash

test() {
    ./test.sh
}

commit() {
}

revert() {
}

test && commit || revert
```

Run script revisited



```
#!/usr/bin/env bash

test() {
    ./test.sh
}

commit() {
    git commit -am working
}

revert() {
}

test && commit || revert
```

Run script revisited

```
#!/usr/bin/env bash

test() {
    ./test.sh
}

commit() {
    if [[ $# -eq 0 ]]
        then git commit -am working
        else git commit -am "$1"
    fi
}

revert() {

}

test && commit "$@" || revert
```

Run script revisited

```
● ● ●

#!/usr/bin/env bash

test() {
    ./test.sh
}

commit() {
    if [[ $# -eq 0 ]]
        then git commit -am working
        else git commit -am "$1"
    fi
}

revert() {
    git reset --hard
    git clean -fd
}

test && commit "$@" || revert
```

Logging?

```
#!/usr/bin/env bash

test() {
    ./test.sh | tee ./tcrfeedback 2>&1
}

commit() {
    # git add .
    if [[ $# -eq 0 ]]
        then git commit -am working
        else git commit -am "$1"
    fi
}

revert() {
    git reset --hard
    git clean -fd
}

test && commit "$@" || revert
```

Git

any
for a bad pull
abuse. verb
mistreat, m
treat badly
e m

Explosion of git history

Copyright Tax



Explosion of git history



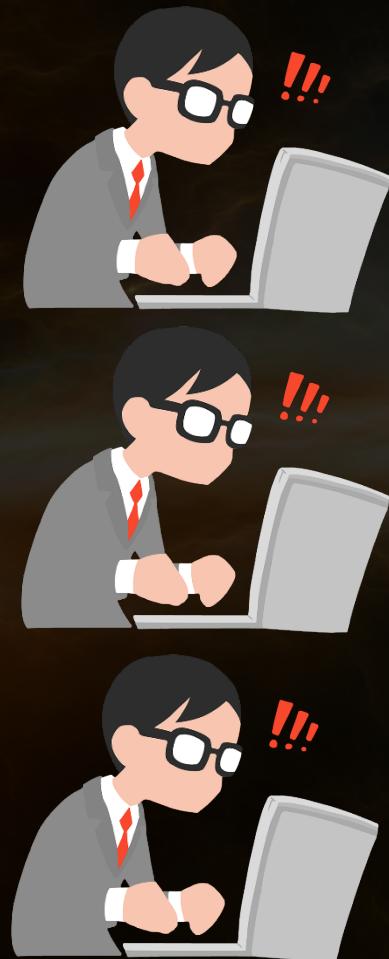
--amend

```
! [rejected]      master -> master (non-fast-forward)
error: failed to push some refs to
```

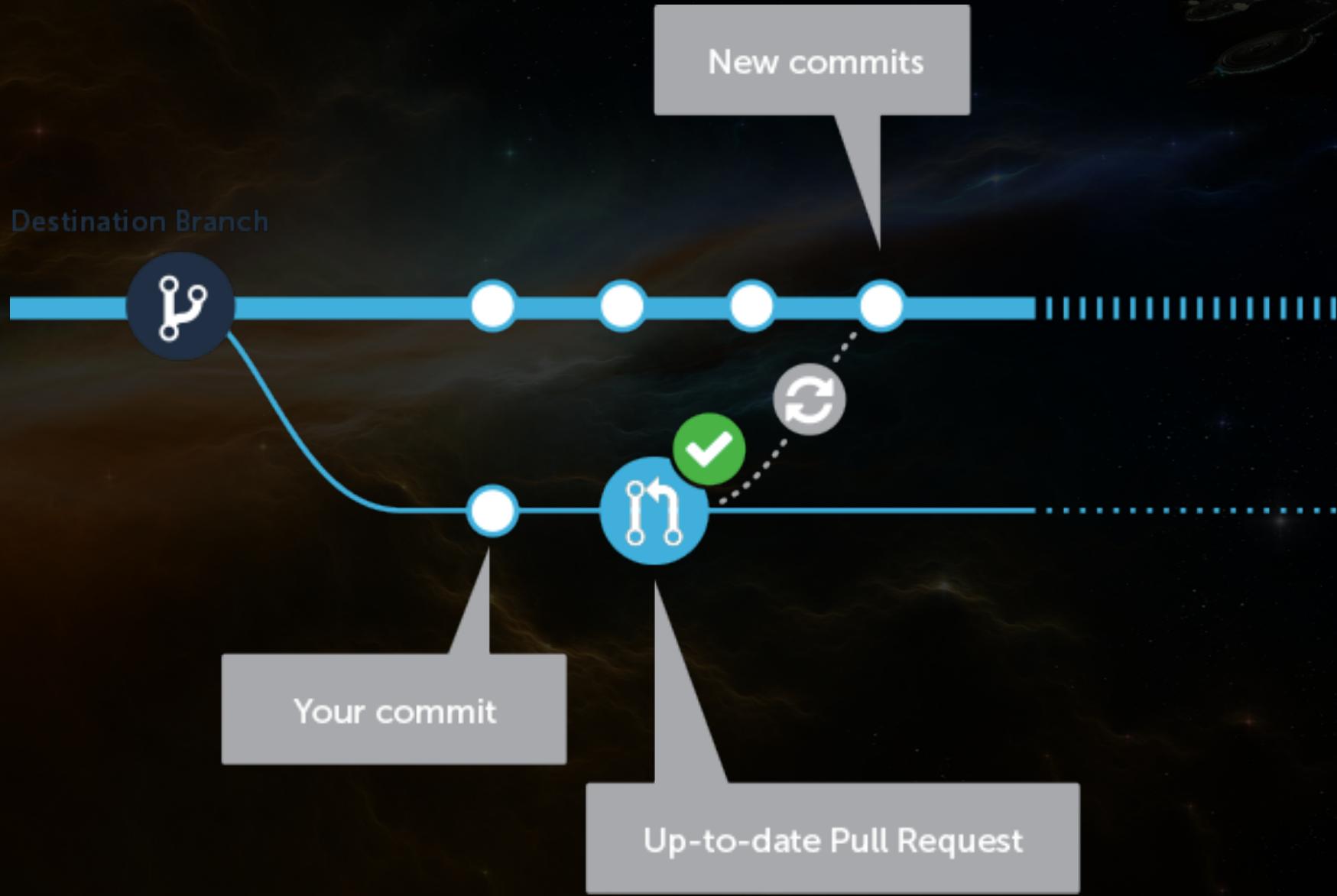
...

```
hint: Updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

One to many

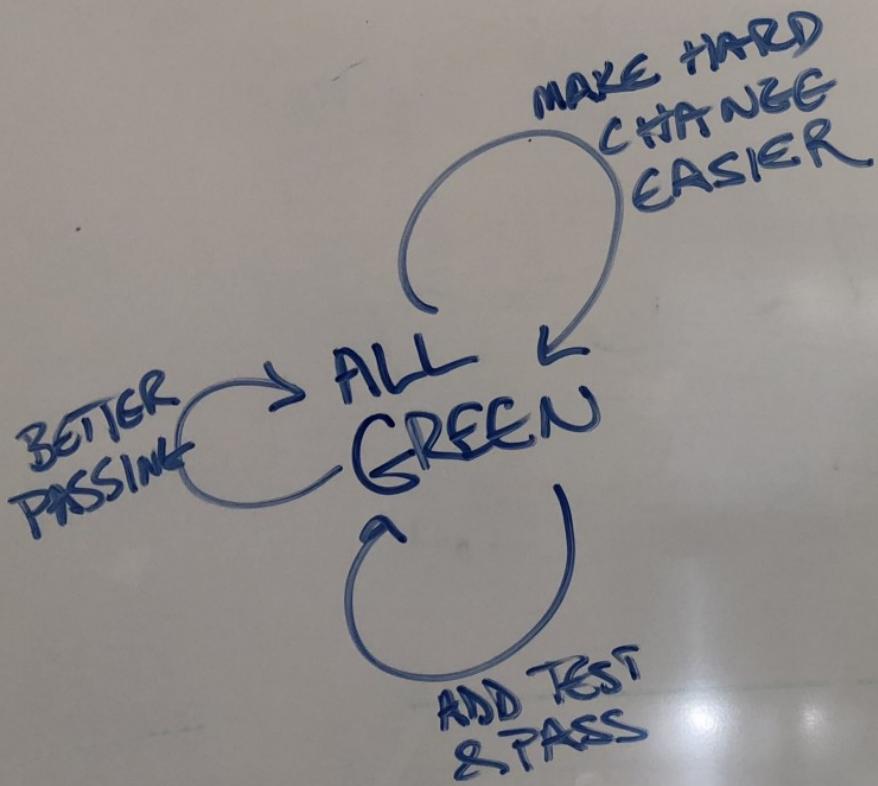


Where is the review part?

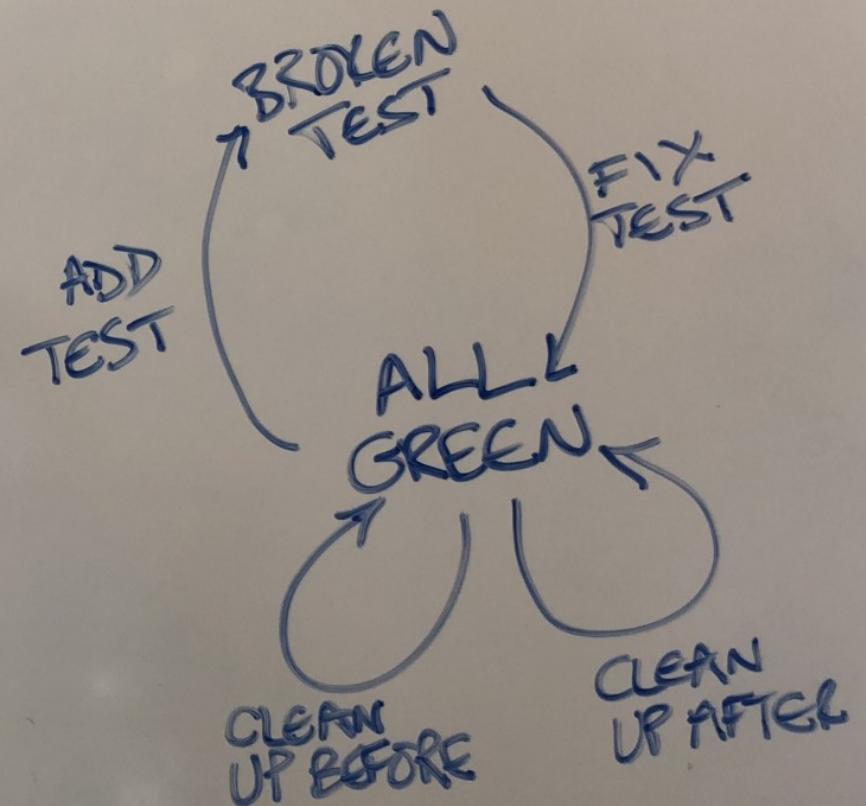


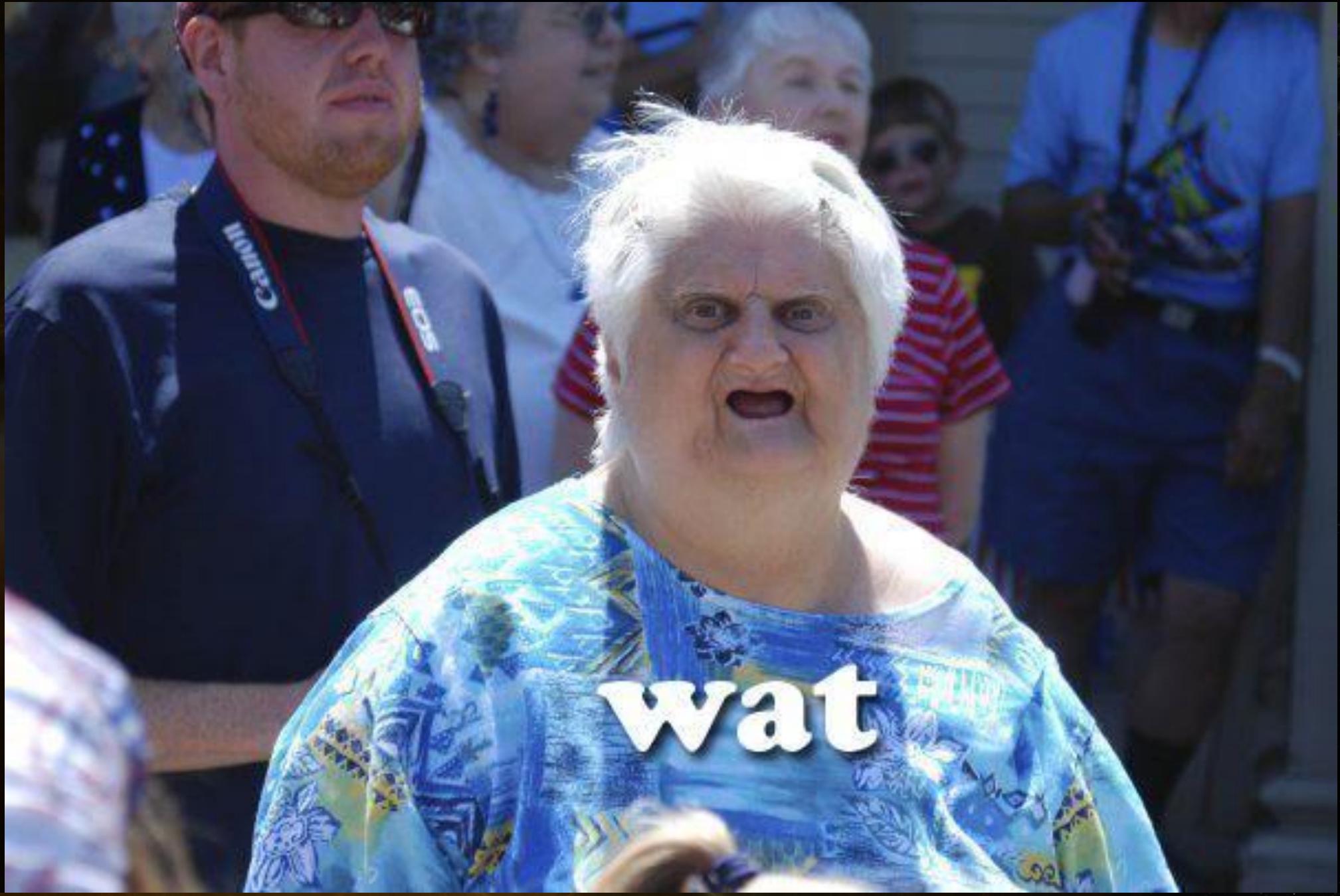
TDD vs TCR

? :



TDD:





Piotr Stawirej | @p_stawirej

Variations

BTCR

The Relaxed/TCTR

The Gentle

The Split

The buddy – Continuous TCR

The watch buddy

The Collaborator

Local Buddy, Remote Team (The Collaborator + The Buddy)



```
if(build().failed)  
    return  
  
if(test().success)  
    commit()  
else  
    revert()
```

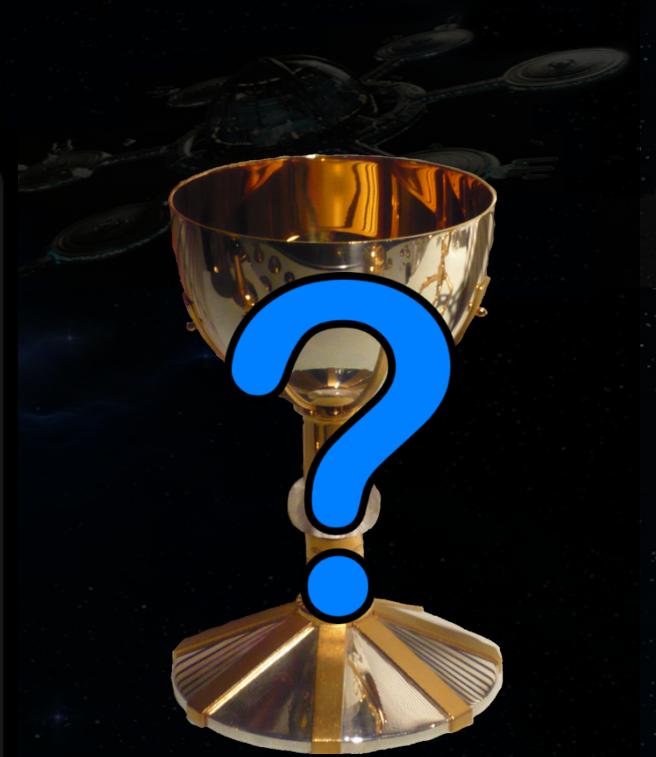


The Relaxed/TCTR



```
if(build().failed)
    return

if(test().success)
    commit()
else
    revert('src/main')
```



The Buddy - Continuous TCR



```
while(true) {  
    tcr()  
}  
  
function tcr() {  
    if(build().failed)  
        return  
  
    if(test().success)  
        commit()  
    else  
        revert()  
}
```

The Watch Buddy

```
while(true) {  
    block_until_change_in_directory('src')  
    tcr()  
}  
  
function tcr() {  
    if(build().failed)  
        return  
  
    if(test().success)  
        commit()  
    else  
        revert()  
}
```

The Storyteller: Beyond the Buddy



Hey, I recognized an error at line 87 at file Source.java, as I executed the Test ‘testFeatureX.java’. To get in a green state ASAP, I delete the lines, that caused the problems. OK?

It seems to work so far. I save the changes and rerun the tests.

Limited WIP

[Reset](#)

Change size watchdog

Enabled

Change size threshold lines

Notification interval minute(s)

Show remaining changes in toolbar

No commits above threshold

Excluded files
e.g. *.approved;**/docs/*.txt

Auto-revert

Enabled

Timeout period minute(s)

Notify on revert

Show time till revert

TCR mode (test && commit || revert)

Enabled

Notify on revert

On passed test

Use commit message from

Don't revert tests

Don't revert files
e.g. *.approved;**/tests/*.js

[» Open readme on GitHub](#)

<https://github.com/dkandalov/limited-wip>

No commits above threshold

Excluded files

`*.approved`



e.g. `*.approved;**/docs/*.txt`

Auto-revert

Enabled



Timeout period

5



minute(s)

Notify on revert



Show time till revert



TCR mode (test && commit || revert)

Enabled



Notify on revert



TDD and TCR



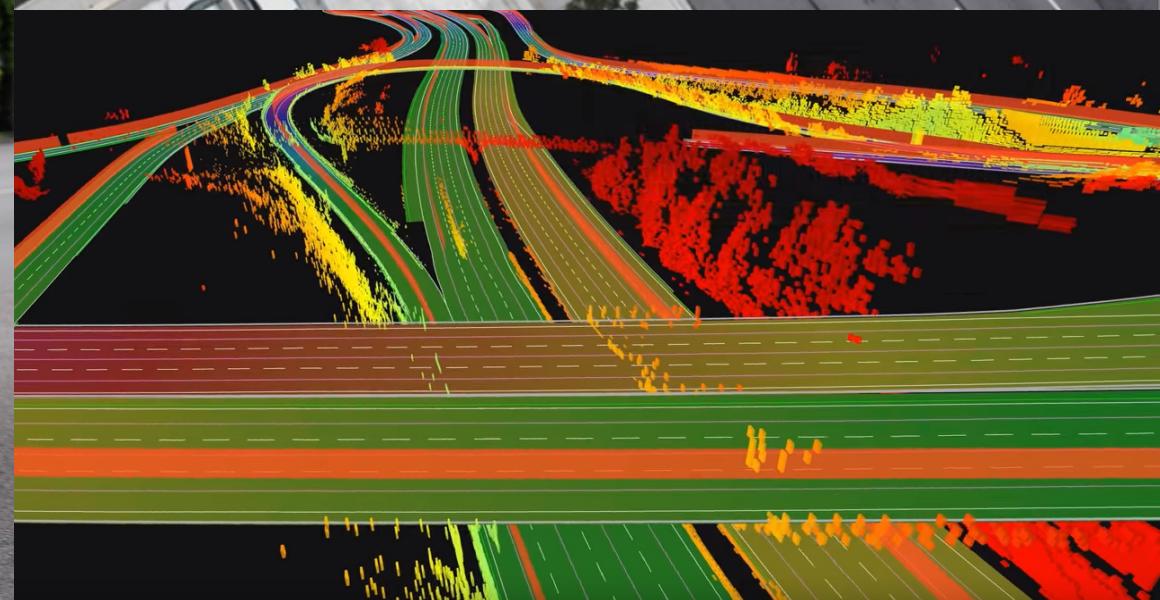
Do we need hybrid solution?

E.g.

```
( test && commit ) || test_code_only || revert
```

WOW, can I use it?

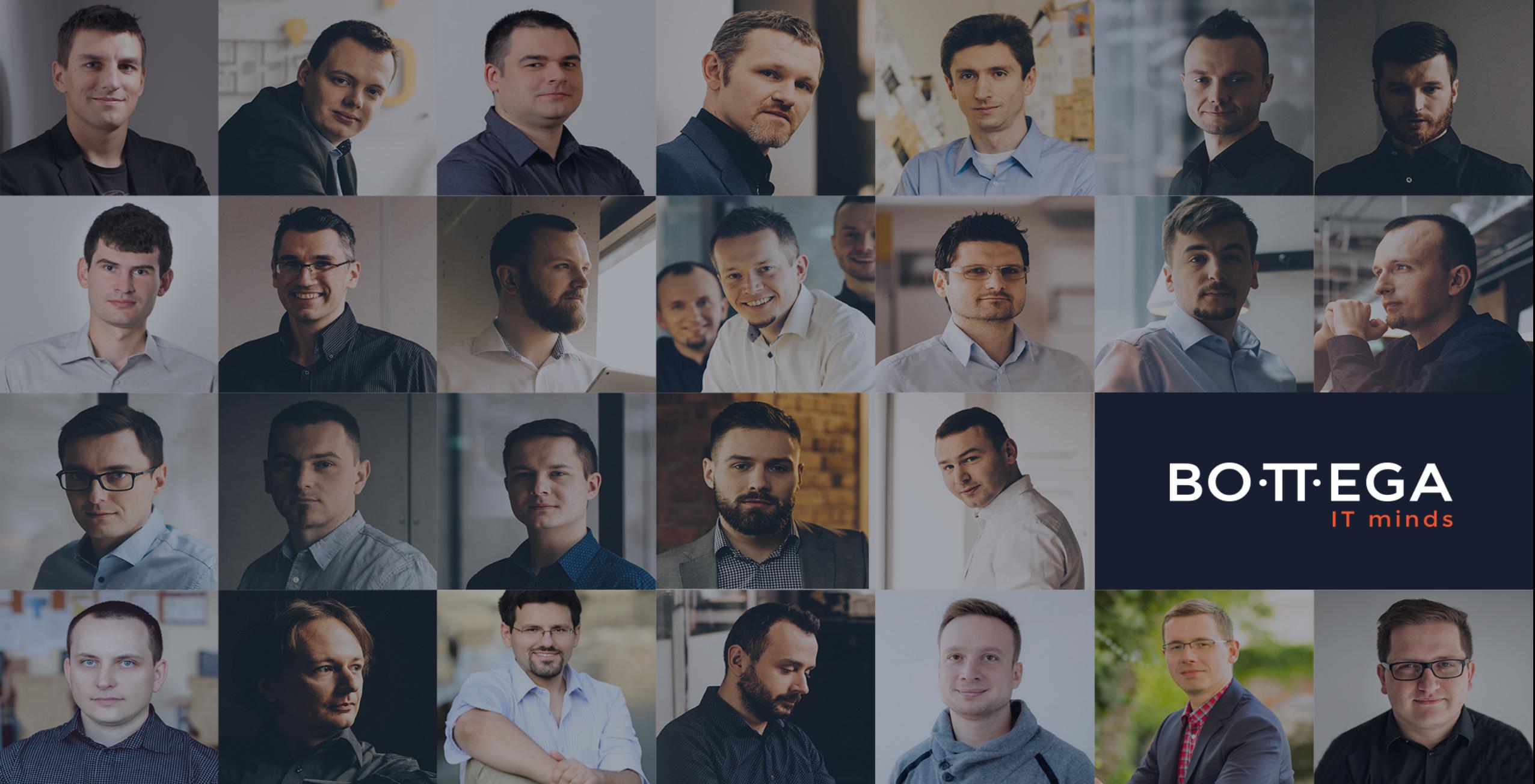




Piotr Stawirej | @p_stawirej



Well yes, but actually no



BO·TT·EGA
IT minds



Takeaways



Alternative workflow.

Small commits.

Cheap to get to known state.

Work/think for minutes, seconds.

Experiment, combine with TDD.

Thank You!

<https://bitbucket.org/piotrstawirej/tcr>

<http://piotrstawirej.bitbucket.io>

@p_stawirej