
Tweak Code Reviews For Maximum Potential

One man's trash (can be) another man's trash



PRAIRIE DEV CON

WEB | DEV | CLOUD | AI



EXCELLENCE IN RECRUITMENT



Survey link:





AJ

Likes: Coding, Dolphins, 3D-Printers
Dislikes: not unsubscribing from observables,
double negatives





CATALIN PIT 💡🚀

@catalinmpit

When somebody pulls your branch and starts making changes to your code 🤢

don't touch my garbage!!!!





What is a code review?

- When you're looking at code you didn't write
 - Meetings (sometimes the meeting is even called: 'Code Review')
 - But could also be in scrum, retrospective, or deployments
 - Paired Programming
 - Ad Hoc 'Hey, do you have a second to look at this?'
 - Pull/Merge Request review
- Can be highly valuable meetings - typically, costly meetings as well
 - High risk, high reward

Tips and Tricks

Hard/Technical

'Easy'/People Skills



- Shift-left (Automate) boring work
- Git Commit Hygiene
- Use templates
- Levelset context/topic
- Micro issues
- Macro issues
- Share Responsibility of resolutions
- Reinforce good practices (not just penalize bad)
- Tone, empathy and ego

Tone, empathy and ego

Tone

- Text communication loses a lot information that exists with in person or video
 - If a PR is racking up comments, meet/call the person
- Asking questions goes a long way



Tone, empathy and ego

Empathy



Tone, empathy and ego

EGO IS THE ENEMY



RYAN HOLIDAY

Reinforce good practices (not just penalize bad)



User Name 3 days ago [↗](#)

Nice work using the container presenter pattern. It looks really clean.

Reply 😊 ...



1

Share Responsibility of resolutions

Could you refactor this to use the **flyweight** pattern while saving in the **memento** pattern?

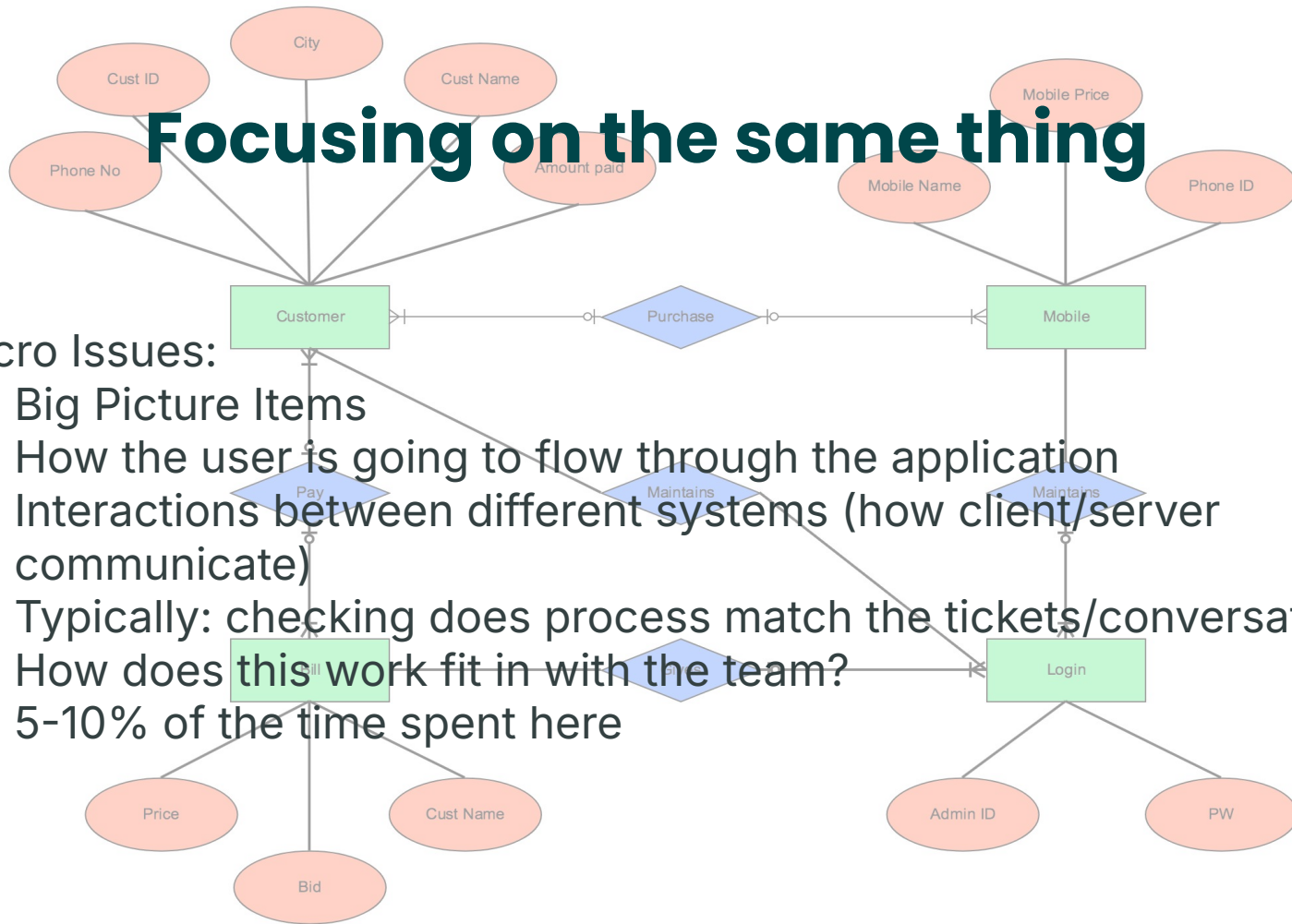
I'm not sure how to do that...

<https://imgtfy.app/?q=flyweight+pattern>



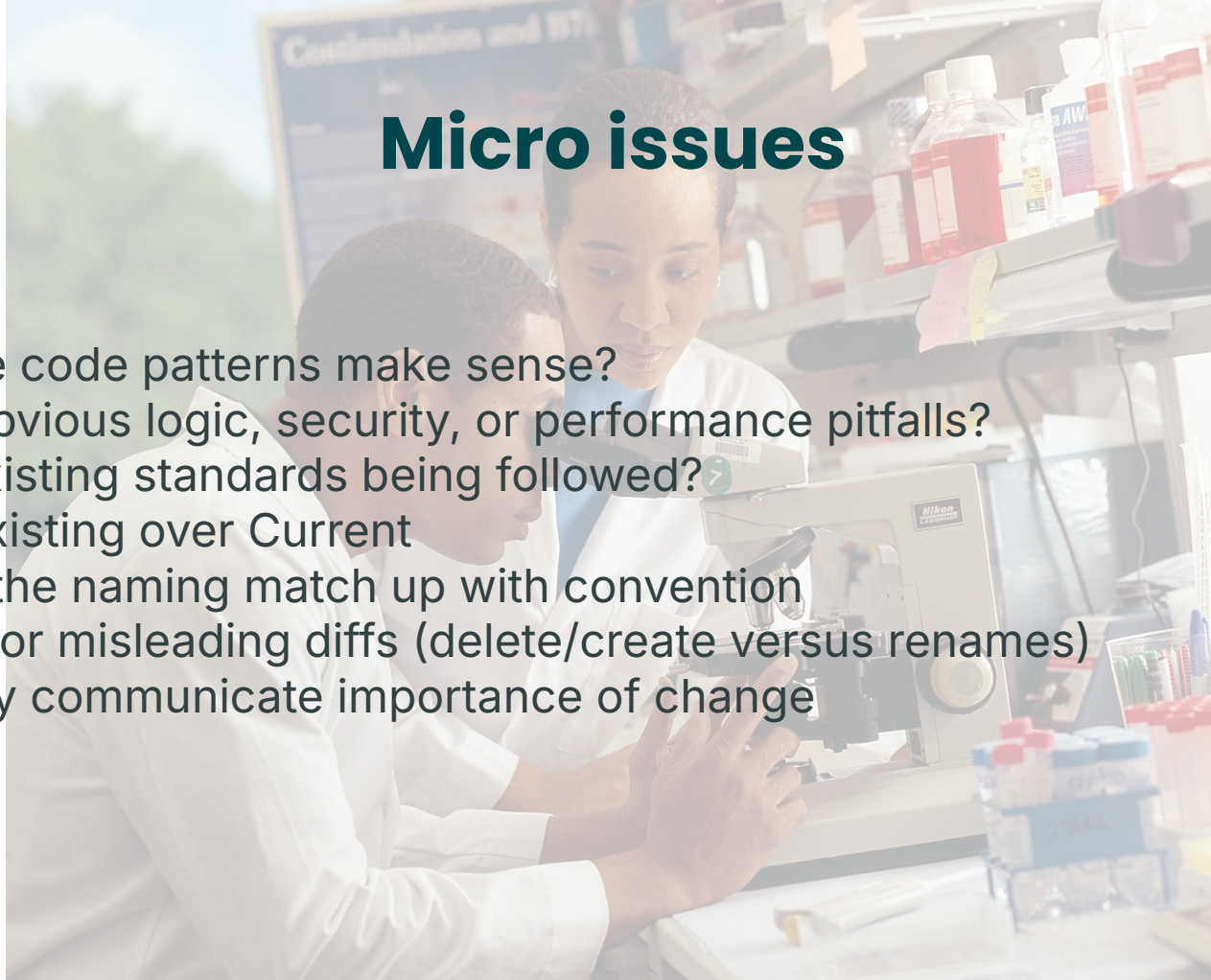
Focusing on the same thing

- Macro Issues:
 - Big Picture Items
 - How the user is going to flow through the application
 - Interactions between different systems (how client/server communicate)
 - Typically: checking does process match the tickets/conversations
 - How does this work fit in with the team?
 - 5-10% of the time spent here



Micro issues

- Do the code patterns make sense?
- Any obvious logic, security, or performance pitfalls?
- Are existing standards being followed?
 - Existing over Current
- Does the naming match up with convention
- Look for misleading diffs (delete/create versus renames)
- Clearly communicate importance of change



Levelset context/topic: le: things both reviewer & creator should know

- Lift and shift work
 - A bunch of green in a PR looks like 'your' work
- PR for a bug in legacy isn't the medium for re-architecting the databases
- Optimization tasks typically take good looking code and make it uglier
- Small similar fixes throughout the app
- Upgrade/Update frameworks

PR Checklist

Please check if your PR fulfills the following requirements:

- ☐ The commit message follows our guidelines: <https://github.com/angular/angular/blob/main/CONTRIBUTING.md#commit>
- ☐ Tests for the changes have been added (for bug fixes / features)
- ☐ Docs have been added / updated (for bug fixes / features)

PR Type

What kind of change does this PR introduce?

- ☐ Bugfix
- ☐ Feature
- ☐ Code style update (formatting, local variables)
- ☐ Refactoring (no functional changes, no api changes)
- ☐ Build related changes
- ☐ CI related changes
- ☐ Documentation content changes
- ☐ angular.io application / infrastructure changes
- ☐ Other... Please describe:

What is the current behavior?

Issue Number: N/A

What is the new behavior?

Does this PR introduce a breaking change?

- ☐ Yes
- ☐ No

Other information

Use Templates – sparingly

https://github.com/angular/angular/blob/main/.github/PULL_REQUEST_TEMPLATE.md

Git Commit Hygiene

```
git commit -m "update the README.md with link to contributing guide"
```

<https://www.conventionalcommits.org/en/v1.0.0/>

```
<type>[optional scope]: <description>
```

```
[optional body]
```

```
[optional footer(s)]
```

<https://gist.github.com/joshbучea/6f47e86d2510bce28f8e7f42ae84c716>

```
feat: add hat wobble
```

```
^__^  ^_____  
|      |  
|      |  
|      +-> Summary in present tense.  
|  
+-----> Type: chore, docs, feat, fix, refactor, style, or test.
```

Git Commit Hygiene

git-rebase-todo

.git > rebase-merge > git-rebase-todo

GitLens Interactive Rebase

Oldest first

EMTN-790-move-formly-blue-component-to-ng-lib-3 Rebasing 10 commits onto ϕ 127de57

Action	Commit Message	Author	Time	Hash
squash	moved the to.class to the theme attribute		4 weeks ago	ϕ 45ac8e3
pick	updated checkbox to use blue field		4 weeks ago	ϕ add023f
reword	updated package name to prefix with @moody's		4 weeks ago	ϕ 8177bac
drop	.npmrc edited online with Bitbucket		last month	ϕ 8b51c2c
pick	.npmrc add auth token to support nodeJS 17+ version		last month	ϕ 1e82c90
reword	Update node label		last month	ϕ 6da3723
edit	update pipeline to handle both projects		4 weeks ago	ϕ 222ede9
pick	update npm to be project aware		4 weeks ago	ϕ 7e7582c
pick	bump the version on the package.json files		last month	ϕ 3a29719
pick	trying to make as similar as possible to emtn-ng wound up fixing when using --no-...		last month	ϕ 1ee7017

Swing file

p Pick r Reword e Edit s Squash f Fixup d Drop alt up Move Up alt down Move Down / Search

DISABLE REBASE EDITOR
WILL ABORT REBASE

SWITCH TO TEXT

START REBASE
CTRL+ENTER

ABORT
CTRL+A

Shift-left (Automate) boring work

- Do naming conventions always lag?
- Does no one read the 458 page coding style guide?
- When conventions change does the project never fully convert?
- Is Typescript leaving you wishing for some added expressivity?
- Is Prettier formatting just not cutting the mustard?

If you've said yes to any of the following:

You might benefit from some Custom ESLint Rules!

Shift-left (Automate) boring work With Custom ESLint Rules

```
You, 4 seconds ago • Uncommitted changes

@Input() label!: string;
value!: string;
warning?: string;

ngOnChanges(changes: SimpleChanges): void {
  if (changes['label'] || changes['value']) {
    if (this.label.length + this.value.length > 50) {
      this.warning = "content must be less than 50 characters";
    } else {
      this.warning = "";
    }
  }
}
```

Custom and Existing ESLint Rules

- Existing ESLint Rules
 - Use existing ESLint Rules to enforce good practices
 - <https://www.bitovi.com/blog/bitovi-eslint-config-handpicked-rules-to-improve-your-code>
- Custom ESLint Rules
 - <https://github.com/bitovi/eslint-plugin>
 - [Make your own ESLint Rule!](#)
 - Uses ASTExplorer for finding something in the code

Conclusion

The background of the slide features a light blue wall with three wooden ladders leaning against it. On the left ladder, a man in a white shirt is climbing upwards. In the center ladder, a woman in a dark dress is climbing. On the right ladder, a man in a white shirt and dark trousers is climbing. The ladders are positioned at different heights, creating a sense of progression and teamwork.

- Pick one thing, try for a sprint, evaluate at retro whether to keep, repeat
- Not every development issue requires a development solution
 - Sometimes talking (video call - not slack) can go a long way
- Think of discussions as being on ladders
 - Closer to the ground - closer to agreement
 - When 'climbing up' the ladder with someone look at ways to de-escalate
 - Make new meeting, agree on proof (Science!), defer to authority, etc
- Overall and most importantly:
- Let curiosity guide you