



Does Your Codebase Spark Joy?

@jennapederson | #NDCOslo

Agenda

- The backstory
- What is joy?
- Why do we want tidy?
- How do we find the magic?



What is the KonMari Method™?

"The KonMari Method™ encourages tidying by category – not by location – beginning with clothes, then moving on to books, papers, komono (miscellaneous items), and, finally, sentimental items. Keep only those things that speak to the heart, and discard items that no longer spark joy. Thank them for their service – then let them go." - konmari.com

What is joy?

Different for each person and each team.

- Small teams - Being productive and confident across the stack
- Larger teams - Being productive and confident across the team and between teams; Being able to act on the risk of a particular change

Define what joy looks like

This is what it looks like for me



Productivity



Confidence



Quality built in



Feedback



Experimentation



Effective decision making



Consistency



Collaboration



Observe and monitor

- 1 We write code for humans
- 2 We're more confident
- 3 We're more productive
- 4 We build quality in early
- 5 Developer onboarding
- 6 Developer happiness
- 7 Cost savings

Why do we want tidy?

More importantly, how do we sell it to management?



We write code for humans

"Programming is the art of telling another human what one wants the computer to do." - Donald Knuth



We're more confident

"One of the magical effects of tidying is confidence in your decision-making capacity." - Marie Kondo



We're more productive

We can find what we need. We can understand it more quickly. We get quicker feedback. We can automate some of our decisions.



We build quality in early

- Because we better understand the code, we're more likely to write higher-quality code
- Bugs are not obscured by complex code
- It becomes easier to test



Developer onboarding

What if your new teammate could on-board to your codebase in one day instead of one month or six months?



Developer happiness

Do your codebase or your development processes make you want to throw things?

@jennapederson | #NDCOslo



Cost savings

All of these lead to cost savings. In the end, this is what the business cares about.

How to find the magic

- 1 Commit to tidying up
- 2 Visualize your destination
- 3 Discard what doesn't spark joy
- 4 Tidy by category, not by location
- 5 Tidy a little everyday

Commit to tidying up

Get management on board and get dedicated time on the calendar.

Visualize your destination

Tidying is not our goal, but it's the tool to get there. We need to define our end state and plan how we'll get there.

Your destination

Improved developer confidence when making changes and committing code

BASELINE

- No test coverage
- No test suite at all
- No continuous integration

HOW YOU'LL GET THERE

- Dedicate one person for entire sprint
- Add test framework to codebase
- Add one passing test
- Set test coverage thresholds to fail the test suite when coverage goes down
- Add CI hook to run test suite on code push
- Set expectations with team
- Task it out and put it on your kanban/sprint board

Your destination

Save time spent on minor nits and common bugs during code reviews

BASELINE

- No code style guide (defined or automated)
- No linters in use
- Comments in code reviews are for formatting style and small or minor errors
- Comments are more critical of the human than the code
- New devs get stuck on syntax and style (with criticism)
- More experienced devs get stuck debating style

HOW YOU'LL GET THERE

- Dedicate one person for entire sprint
- Use a standard style guide for your stack
- Implement a linter on save
- Implement a code formatter on save
- Run each one time on the entire codebase
- Set expectations with team
- Task it out and put it on your kanban/sprint board

Discard what doesn't spark joy

Don't skip this step. Dead code. Dead features.
Unhelpful teammate. Unmaintained tests. Data. Heavy
process.

Tidy by category, not by location

Tidy in the right order.

Areas to tidy

- 1 Code formatting & style
- 2 Common errors
- 3 Testing
- 4 Names & comments
- 5 Refactoring
- 6 Beyond your codebase



Code formatting & style

Using an automated formatter eliminates nits in code review comments and helps newcomers

Example: Prettier



Common errors

Using a linter helps find and fix common programming mistakes like deprecated APIs, unused variables or functions, undefined variable assignments, and security vulnerabilities

Example: ESLint



Testing

Having a well-maintained, passing test suite allows you to confidently change your code

Examples: Unit tests, end-to-end tests



Names & comments

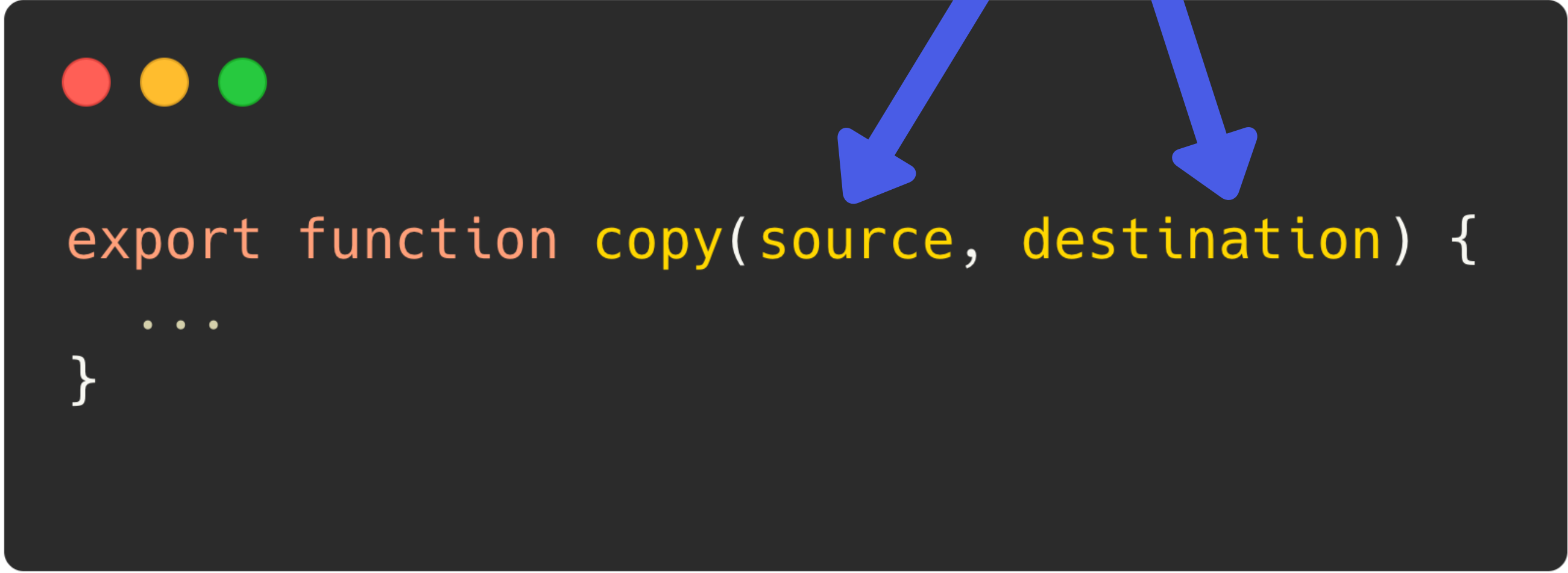
Both names and comments communicate your intent to your reader

Don't do this



```
export function copy(x1, x2) {  
  ...  
}
```

Do this instead



```
export function copy(source, destination) {  
  ...  
}
```


Don't do this

```

● ● ●
// a user can be reimbursed if they are an employee that can expense
// receipts and they started more than 90 days ago
if (user.isEmployee & user.canExpense &&
    moment().diff(moment(user.startDate), 'days') >= 90) {
    ...
}

```

Do this instead



```
if (isEligibleForReimbursement(user)) {  
    ...  
}
```



Refactoring

Take a series of small structural changes, supported by tests to make your code easier to change

Feathers' Algorithm

For refactoring legacy code

- 1 Identify Change Points
- 2 Find test points
- 3 Break dependencies
- 4 Write tests
- 5 Make changes and refactor



Beyond your codebase

- Monitoring & observability
- UX & design
- Security
- Tools & processes

Tidy a little everyday

Change your mindset. Make it a habit.

@jennapederson | #NDCOslo

1

Train people

2

Set expectations

3

Have a plan

4

Reserve time for tidying

Make it a habit



Train people

- Learn the tools
- Learn software design principles
- Learn the refactoring techniques
- Start a book club
- PRACTICE



Set expectations

- Style guides
- Test coverage
- Broken builds
- Always leave the codebase cleaner than you found it




Have a plan

- Plan it out like you would a new feature or project
- Task it out
- Put it on your kanban/sprint board
- Set goals and track to them
- Make progress visible



Reserve time for tidying

- Build time into your sprints, whether it's 25% of one person or 100% of the team
- Do this every sprint



"If you tidy up in one shot, rather than little by little, you can dramatically change your mind-set."

- Marie Kondo

@jennapederson | #NDCOslo

**Define your joy.
Find your magic.**

**Clearing the ~~physical~~ digital clutter will
help you clear the mental clutter.**

Thank you!

Q & A

@jennapederson | #NDCOslo

