

WHO NEEDS CODE QUALITY ANYWAY?



A WordPress Perspective



About me

I am a WordPress and PHP specialist

I started my first projects in the late 90s.

I worked on projects with millions of page impressions. As technical lead I developed custom themes, plugins, optimized performance and planned coming features.

I am a frequent speaker at WordPress events, and founder and co-host of the Meetup in Dortmund.

Currently I work as Senior Web Developer, WordPress Expert and DX Engineer at Coding Pioneers in Iserlohn.

<https://www.linkedin.com/in/christoph-daum/>



Material at the end of the talk

- Full slides available
- GitHub Repository with hundreds of examples
- Questions



Meet Sheldon

- He mostly worked alone
- He is confident of his skills
- He likes how WordPress is coded

Hi, I'm Sheldon



Q: Coding Standards

*Applying coding standards
doesn't improve my code, it's just a
waste of time.*



Q: Coding Standards

*And don't forget.
"Code is Poetry"!*



A: Coding Standards

Coding is not an Art - Coding is a Craft

- Coding Standards help to work together
- Unified Styling (Usages of spaces, naming conventions etc.)
- Navigating through code will become easier
- Technical sniffs
 - Escaping
 - Proper preparation for i18n
 - Warning that meta queries are slow



Q: PHPDoc

*The code is documentation
enough, the Coding Standards want
me to add PHPDoc. That is not
worth the time.*



A: PHPDoc

- AI reads PHPDoc and thus will provide better suggestions
- Your IDE reads the Docs and your auto complete will be better
- Describe what a function or class should do, this helps finding bugs
- Can help others to understand how a function is used
- Can show when a function was added, changed and more



Q: Professional Comments

*Everyone likes a good
pun or banter in a comment.
I always speak my mind...
or rather write it.*



Q: Professional Comments

And it's just a private project, no one will ever read my comments



A: Professional Comments

- Build a habit of writing professional comments
- Comments can be read by colleagues, clients, future employers
- Ranting, insulting or other offensive comments can cost your reputation
- Comments are not for venting, insulting and more
- If a comment is on a public github repository, everyone can read it

Found in the JavaScript of an actual project

```
// I'm very sorry. It had to be done. Quick, hard and dirty,  
// that's how you like it, isn't it...
```



Q: Commented out code

*I will remember
why I left the commented out code
there.*



A: Commented out code

- Always write when and why code was commented out
- If you forget to remove it, it will be kept forever
- In most cases, your git history is all you need
- Better write a comment with a hint to your git repository



Q: Commits & Commit Messages

*Well good luck with that,
I only commit once a week, and I
never know what to use as a
commit message.*



Q: Commits & Commit Messages

*I guess you will tell me
"Misc"
is not a good message?*



A: Commits & Commit Messages

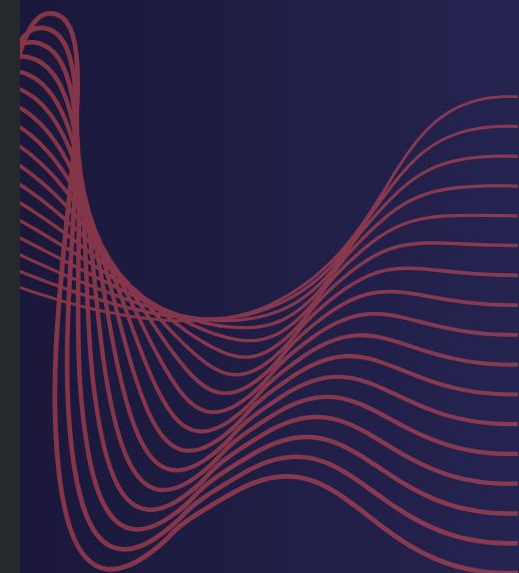
- Your commit messages should also follow guidelines
- For example “Conventional Commits”
- Commit regularly
 - Only one single task/feature or fix per commit
 - Never break the project, the project should run at any given commit
- Document properly
 - Keep the summary under 50 characters
 - Wrap body text at 72 characters (50/72 Rule)
 - Explain why and not just what
 - Link your ticket if available
 - Anyone should be able to understand what happened in a commit, just by reading the commit message
- Let AI help you!

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



A: Commits & Commit Messages

readme	Sheldon	0d9361d	7. Dec 2016 at 07:42
css changes	Sheldon	feee1ec	7. Dec 2016 at 07:32
readme	Sheldon	c786942	7. Dec 2016 at 07:16
css tweaks	Sheldon	46d5f35	7. Dec 2016 at 07:14
v1.1.1 Version 1.1.1 * fixed keyboard changed	Sheldon	327578f	7. Dec 2016 at 07:10
v1.1.0 new version 1.1.0, watermark, statusbox, keyboard shortcuts	Sheldon	d21f12a	6. Dec 2016 at 22:49
v1.0.0 missing version bump fixed	Sheldon	dea29e0	29. Oct 2016 at 22:46
1.0.0	Sheldon	4d564b3	28. Oct 2016 at 22:52
v0.9.11 update to 0.9.11	Sheldon	051d0fe	28. Oct 2016 at 07:50
v0.9.10 Update README.md	Sheldon	aaf014b	26. Oct 2016 at 15:27
Update README.md ↵	Sheldon	2d4665c	26. Oct 2016 at 15:25
Update README.md	Sheldon	b2495d7	26. Oct 2016 at 15:15
Update apermo-adminbar.php ↵	Sheldon	f5174bd	26. Oct 2016 at 15:14
v0.9.9 Create script.js	Sheldon	f216bd0	22. Oct 2016 at 09:07
Add files via upload	Sheldon	8c73c10	22. Oct 2016 at 09:06
Update apermo-adminbar.php ↵	Sheldon	e3dec79	7. Jul 2016 at 22:19
compatibility with domain mapping for multisite	Sheldon	54356b4	1. Jul 2016 at 08:38
v0.9.5 fixed bug on subfolder installations	Sheldon	d77fbc6	30. Jun 2016 at 17:57
v0.9.4 export & import	Sheldon	de61a0d	30. Jun 2016 at 08:08
typos	Sheldon	3cab6d1	29. Jun 2016 at 22:52
v0.9.3 added new filter, removed scheme url from options	Sheldon	625f541	29. Jun 2016 at 22:39
v0.9.2 Merge pull request #1 from 2ndkauboy/master ↵	Sheldon	c2678cf	29. Jun 2016 at 08:04
some minor code improvements and removing the readme.txt file which is no longer need...	2ndkauboy	0eeb464	28. Jun 2016 at 23:42
v0.9.1 Bug fixes	Sheldon	6d44122	28. Jun 2016 at 21:02
v0.9.0 First release	Sheldon	00f89cc	28. Jun 2016 at 19:20
Initial commit	Sheldon	316ef5d	28. Jun 2016 at 18:46



A: Commits & Commit Messages

✓ release/1.4.0		fix: remove indie-action web actions ↻	👤 Christoph Daum	75a513a	26. Mar 2026 at 13:35
•	test:	update tests for microdata removal ↻	👤 Christoph Daum	c445a71	26. Mar 2026 at 12:41
•	refactor(images):	remove microdata ↻	👤 Christoph Daum	254a3b6	26. Mar 2026 at 09:57
•	refactor(templates):	remove microdata ↻	👤 Christoph Daum	e475a51	26. Mar 2026 at 09:55
•	refactor(comment):	remove microdata ↻	👤 Christoph Daum	e85b8c1	26. Mar 2026 at 09:48
•	refactor(tags):	remove microdata ↻	👤 Christoph Daum	c2b13c8	26. Mar 2026 at 09:47
•	refactor(semantics):	remove microdata ↻	👤 Christoph Daum	1403470	26. Mar 2026 at 09:38
•	test(schema):	add unit tests ↻	👤 Christoph Daum	f163f00	26. Mar 2026 at 09:34
•	feat(schema):	add Yoast SEO integration ↻	👤 Christoph Daum	4a78852	26. Mar 2026 at 09:11
•	feat(schema):	add JSON-LD structured data ↻	👤 Christoph Daum	f238099	26. Mar 2026 at 09:09
•	fix:	correct 500.php template description ↻	👤 Christoph Daum	10aa996	26. Mar 2026 at 08:15
•	fix:	remove dead wp_localize_script call ↻	👤 Christoph Daum	b908e4c	26. Mar 2026 at 08:14
•	fix:	replace bloginfo() with escaped output ↻	👤 Christoph Daum	14f4d35	26. Mar 2026 at 08:14
•	fix:	remove type="text/css" from style tag ↻	👤 Christoph Daum	90846e3	26. Mar 2026 at 08:13
•	fix:	remove wp_body_open() compat shim ↻	👤 Christoph Daum	42faec3	26. Mar 2026 at 08:12
•	fix:	remove get_self_link() polyfill ↻	👤 Christoph Daum	168a8ce	26. Mar 2026 at 08:10
✓	origin/main	v1.3.0 Merge pull request #46 from apermo/release/1.3.0 ↻	👤 Christoph Daum	0e6d197	22. Mar 2026 at 22:14
•	release/1.3.0	fix: clean up the_content method ↻	👤 Christoph Daum	4a63b25	22. Mar 2026 at 22:06
•	fix:	use @see for WP core hook docblocks ↻	👤 Christoph Daum	0f7b836	22. Mar 2026 at 21:42
•	fix:	keep named parameters for get_the_content ↻	👤 Christoph Daum	f180e9f	22. Mar 2026 at 21:27
•	fix:	address PR review feedback (round 4) ↻	👤 Christoph Daum	5a278d9	22. Mar 2026 at 21:18
•	fix:	address PR review feedback (round 3) ↻	👤 Christoph Daum	9f02aae	22. Mar 2026 at 21:05
•	docs:	update README, CHANGELOG, CLAUDE.md for 1.3.0 ↻	👤 Christoph Daum	f807919	22. Mar 2026 at 20:06
•	chore:	remove unused entry-nav template part ↻	👤 Christoph Daum	075df37	22. Mar 2026 at 19:35
•	test:	expand E2E suite to 31 tests (#49) ↻	👤 Christoph Daum	fee01c8	22. Mar 2026 at 19:17
•	docs:	add 1.3.0 changelog entry	👤 Christoph Daum	946a257	22. Mar 2026 at 18:01



Q: Main Branch vs feature branches

*Working directly on
the main branch is faster and easier.
Feature branches are overhead and I
don't have enough time
for that.*



A: Main Branch vs feature branches

Will cause problems if...

- you work with multiple people
- you work on several features at once
- you want to do major refactoring is basically impossible
- you want to have multiple stages of your website



Q: Pull Request and Code Review

Ok, you have a point, but I will just merge my changes into main once they are ready. Doing code review is still too slow.



A: Pull Request and Code Review

- Code Review is a main staple in software development
- Share and learn from each others
- 4 eyes principle for code
- Prevent logic errors
- Prevent Debug Code from being merged
- Fix Typos
- Gain confidence before deploying



Q: Pair Programming

*If a second developer
should look at my code, I heard
about pair programming.
What about that?*



A: Pair Programming

- Pair Programming CAN replace Code Review - depending on your teams rules
- for sharing knowledge
- for debugging
- for developing ideas
- for software architecture



Q: AI Review

*Ok, but I sure can let Claude,
Gemini or CoPilot review my changes,
that's a review.*



A: AI Review

Is a good idea... **BUT**

- Might not know the full Context
- Will miss things
- Does not take away the responsibility

In our Code Review process it became an anchor in the last months, first review by multiple AIs, followed by a human review



Q: PHP7&8 Type Hints

One of the strengths of PHP is being type less. I don't need to add types, that just slows me down.



A: PHP7&8 Type Hints

- Strict Typing helps to make your code more robust and predictable
- Errors will occur visibly instead of being weird side effects
- Can improve security
- Clear contract what is delivered and what is received
- Improves readability
- Can replace/simplify validation and sanitization
- Simplifies testing scenarios



Q: Strict comparisons and conditions

*Checking if a value is
truthy or falsy is enough, strict
comparisons are tedious.*



A: Strict comparisons and conditions

Strict comparison in conditions will prevent unpredicted behaviour

Instead of checking if something is truthy, you can check if contains the right type.

```
if ( $post ) {  
    echo $post->post_title;  
}
```

```
if ( $post instanceof WP_Post ) {  
    echo $post->post_title;  
}
```

Can result in a fatal error if \$post is WP_Error



Q: DRY - Don't repeat yourself

*Copy and paste is faster
than refactoring.
And I will just do it once!
I promise! Really!!*



A: DRY - Don't repeat yourself

Copy-pasting creates maintenance nightmares.

- ✓ Extract repeated logic
- ✓ Create reusable functions
- ✓ Share common validation
- ✓ Use inheritance/composition wisely
- ✗ Don't copy-paste code
- ✗ Don't repeat validation
- ✗ Don't duplicate queries
- ✗ Don't over-abstract simple logic
- ✗ Don't create unnecessary dependencies



Q: KISS - Keep it simple, stupid

*I want to show my skills,
it won't hurt if I overengineer
this prototype.
The client will LOVE it.*



A: KISS - Keep it simple stupid

✓ Choose simple solutions

✓ Avoid unnecessary complexity

✓ Write readable code

✓ Use familiar patterns

✓ Question every abstraction

✗ Don't use design patterns for simple tasks

✗ Don't add layers without reason

✗ Don't make simple things complicated

✗ Don't prioritize "clever" over "clear"



Q: YAGNI - You ain't gonna need it

*If my plugin is a success,
I will release this to TYPO3, Joomla
and Drupal. I'll directly prepare my
code for this.*



A: YAGNI - You ain't gonna need it

✓ Build what you need NOW

✓ Add features WHEN requested

✓ Keep it simple

✓ Refactor when pattern emerges

✗ Don't build "just in case"

✗ Don't over-engineer

✗ Don't add speculative features

✗ Don't create unused abstractions

✗ Don't skip all planning



Q: Software Design Principles (SOLID)

Ok, but a function or class that will do everything is easier to read and you can follow the code.



A: Software Design Principles (SOLID)

- **S**ingle Responsibility
- **O**pen-closed principle
- **L**iskov substitution principle
- **I**nterface segregation principle
- **D**ependency inversion principle



<https://en.wikipedia.org/wiki/SOLID>



A: Single Responsibility

Each class and function should only have one responsibility

- Maintainability - Simple and well-defined code is easier to understand and modify
- Testability - It's easier to test, especially with Unit Test
- Flexibility - Changes to one part won't have side effects to other parts



A: Open-closed principle

Code should be open for extension but closed for modification

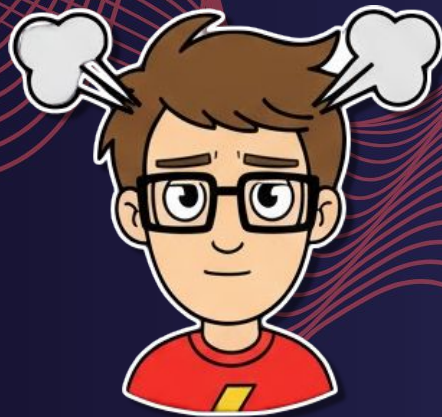
- Extensibility - new features can be added without modifying existing code
- Stability - reduces the risk of introducing bugs when making changes
- Flexibility - adapts to changing requirements more easily



A: Liskov substitution principle

Code must be able to use objects of derived classes without knowing it

- Polymorphism - Making code more flexible and reusable.
- Reliability - Subclasses adhere to the contract defined by the superclass.
- Predictability - Using the object of a subclass won't break the program



A: Interface segregation principle

Code should not depend upon interfaces that it does not use.

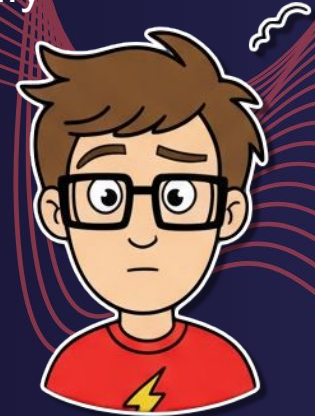
- Decoupling - Reduce dependencies between classes, make the code more modular and maintainable.
- Flexibility - Allow for more targeted implementations of interfaces.
- Avoids unnecessary dependencies - Don't have to depend on methods you don't use.



A: Dependency inversion principle

Depend upon abstractions, not concretes.

- Loose coupling - Reduce dependencies between modules, make the code more flexible and easier to test
- Flexibility - Enable changes to implementations without affecting clients
- Maintainability - Makes code easier to understand and modify



Q: Testable code

Ok, ok.

*But why write testable code. I don't
do test driven development and I
need to ship these changes.*



A: Testable code

Testable code or code following SOLID, is also good without test

- Simple functions are easy to understand
- Bugs are easy to find or prevent
- Side effects are rare
- Tests can be added at a later point



Q: AI does it for me

*I will be vibe coding everything.
So I don't need to
learn all of that.*



A: AI does it for me

That's a topic for a whole new talk.

In Short: You still need to be able to review, understand and question what AI does.



Q: A11y, I18n, SEO and more

*And what about A11y, I18n,
and SEO, do they benefit from code
quality, too?*



A: AIly, I18n, SEO and more

In short words

YES

But I'm out of time, so I
can't cover it today



Share knowledge and learn from peers

*Ok, I think I learned a
thing or two...
maybe I should share it,
like you do...*



Bonus Tip 1: PHP Inspections EA Extended

Static Code Analysis tool for PhpStorm (sorry VS Code People)

Offers suggestions for

- Performance
- Simplification
- Modernization
- Security

The background features a series of overlapping, wavy lines in shades of red and orange, creating a sense of motion and depth. In the center-right, there is a large, multi-faceted graphic that resembles a gemstone or a stylized letter 'W'. This graphic is primarily magenta with purple and blue facets. Inside this graphic, the text 'PHP Inspections' is written in a bold, white, sans-serif font, with 'EA Extended' written below it in a smaller, white, sans-serif font.

**PHP
Inspections**
EA Extended



Bonus Tip 2: Composer & Autoloader

WordPress and Composer do work together

For themes and Plugins

- Autoloader
- Packages
- Coding Standards
- And much more

For Projects

- Bedrock <https://roots.io/bedrock/>



Minecraft?



Bonus Tip 3: Coding Standards

Implementing PHPStan, eslint and co is easier than ever before.

The excuse “we don’t have the time for that” is no more. Adding automated coding standard fixes on commit or via github action is now done quicker than a few manual reviews.

And it’s also super easy to write your own sniffs now.

<https://github.com/apermo/apermo-coding-standards>



Bonus Tip 4: PHPStan

Implementing PHPStan and similar static analysis tools is easier than ever before.

The excuse “we don’t have the time for that” is no ... oh wait i have a déjà-vu

<https://github.com/apermo/apermo-phpstan-wordpress-rules>



Bonus Tip 5: Playwright, PHPUnit, Codecov

As you already guessed, adding more automated tests is just as easy as the others.

For my projects I added any automated test that made sense to me, in order to test everything before another developer (or myself in case of agentic coding) spends any time on a review.

I created a template for my WordPress projects.

<https://github.com/apermo/template-wordpress/>



Bonus Tip 6: [ANYTHING YOU ALWAYS WANTED]

- Automated deployment
- Simple sync of database from prod to local
- Finally use proper documented issues or tickets.



Share knowledge and learn from peers

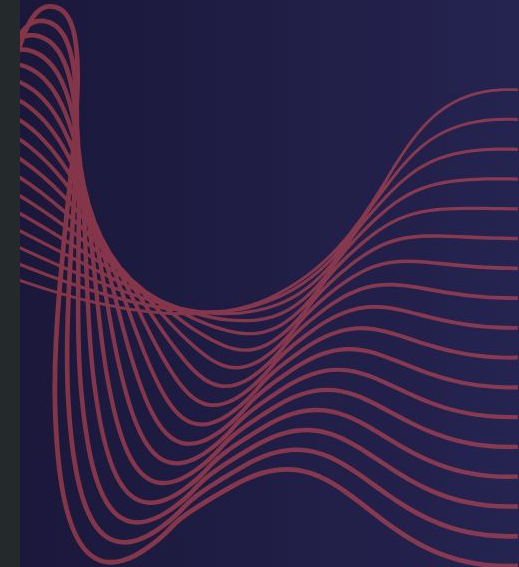
- Sharing and learning is important
- Writing good code will also help others to learn
- Doing Code Review does improve your own skills and your peer's skills

But I have one more thing...



Sorry I lied

readme	Christoph	0d9361d	7. Dec 2016 at 07:42
css changes	Christoph	feee1ec	7. Dec 2016 at 07:32
readme	Christoph	c786942	7. Dec 2016 at 07:16
css tweaks	Christoph	46d5f35	7. Dec 2016 at 07:14
v1.1.1 Version 1.1.1 * fixed keyboard changed	Christoph	327578f	7. Dec 2016 at 07:10
v1.1.0 new version 1.1.0, watermark, statusbox, keyboard shortcuts	Christoph	d21f12a	6. Dec 2016 at 22:49
v1.0.0 missing version bump fixed	Christoph	dea29e0	29. Oct 2016 at 22:46
1.0.0	Christoph	4d564b3	28. Oct 2016 at 22:52
v0.9.11 update to 0.9.11	Christoph	051d0fe	28. Oct 2016 at 07:50
v0.9.10 Update README.md	Christoph Daum	aaf014b	26. Oct 2016 at 15:27
Update README.md ↔	Christoph Daum	2d4665c	26. Oct 2016 at 15:25
Update README.md	Christoph Daum	b2495d7	26. Oct 2016 at 15:15
Update apermo-adminbar.php ↔	Christoph Daum	f5174bd	26. Oct 2016 at 15:14
v0.9.9 Create script.js	Christoph Daum	f216bd0	22. Oct 2016 at 09:07
Add files via upload	Christoph Daum	8c73c10	22. Oct 2016 at 09:06
Update apermo-adminbar.php ↔	Christoph Daum	e3dec79	7. Jul 2016 at 22:19
compatibility with domain mapping for multisite	Christoph	54356b4	1. Jul 2016 at 08:38
v0.9.5 fixed bug on subfolder installations	Christoph	d77fbc6	30. Jun 2016 at 17:57
v0.9.4 export & import	Christoph	de61a0d	30. Jun 2016 at 08:08
typos	Christoph	3cab6d1	29. Jun 2016 at 22:52
v0.9.3 added new filter, removed scheme url from options	Christoph	625f541	29. Jun 2016 at 22:39
v0.9.2 Merge pull request #1 from 2ndkauboy/master ↔	Christoph Daum	c2678cf	29. Jun 2016 at 08:04
some minor code improvements and removing the readme.txt file which is no longer need...	2ndkauboy	0eeb464	28. Jun 2016 at 23:42
v0.9.1 Bug fixes	Christoph	6d44122	28. Jun 2016 at 21:02
v0.9.0 First release	Christoph	00f89cc	28. Jun 2016 at 19:20
Initial commit	Christoph Daum	316ef5d	28. Jun 2016 at 18:46



I am Sheldon... we all are.



~~\$\$\$~~

WHO NEEDS

CODE

QUALITY

ANYWAY?!



Questions?
Thank you for your attention!



Slides at:
c13s.com/wordcamp



Further examples:
github.com/apermo

linkedin.com/in/christoph-daum/

