

Lessons from **WordPress** core  
on how to *(not)* write code

You can't **afford** WordPress code

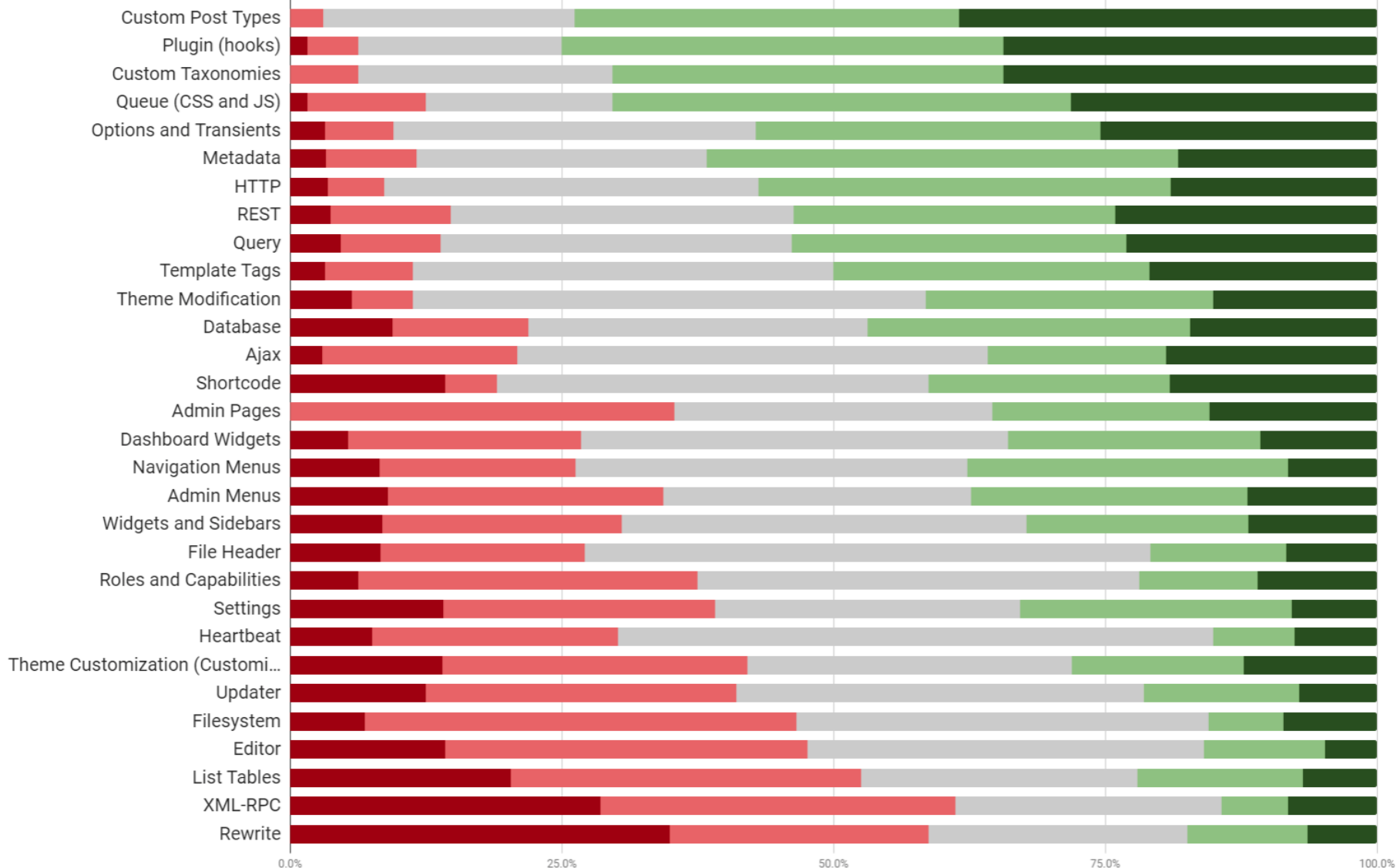
# WordPress core development

1. technical debt → popularity
2. popularity → volunteers
3. volunteers → technical debt

# WordPress in StackOverflow survey

- 6th Most Dreaded Platform (2018)
- 3rd Most Dreaded Platform (2017)
- 2nd Most Dreaded Technology (2016)
- 3rd Most Dreaded Technology (2015)

Not “is it good?”  
but “**what** is good?”



# Best APIs

1. Custom Post Types (74% positive)
2. Plugin (hooks) (75%)
3. Custom Taxonomies (70%)
4. Queue (CSS and JS) (70%)
5. Options and Transients (57%)

# Best APIs – commonalities

- laconic
- procedural
- flexible
- public



# Worst APIs

1. Rewrite (59% negative)
2. XML-RPC (61%)
3. List Tables (53%)
4. Editor (48%)
5. Filesystem (47%)

# Worst APIs – commonalities

- verbose
- persistent
- clunky
- semi-private

**WordPress is a **story****  
**not a blueprint**

# Be WordPress – good, not WordPress – bad

- make great APIs
- hold on to your wins
- part with your failures
- love users *and* developers
- write reliable flexible code

# Andrey “Rarst” Savchenko

- [Rarst.net](https://rarst.net)
- WordPress contractor
- Date/Time core component maintainer

[Rarst.net/slides/core-lessons](https://rarst.net/slides/core-lessons)