

# Technical Debt Status Check for Laravel Developers

Assessing the scale of your technical debt, and the effort it would take to clean it up, isn't always a straightforward task. If you need to back up your argument for technical debt repayment, start with a self-assessment of your codebase's status.

Get started by answering each question and adding up your score. Then, get a final rating that will help you plan how to approach a repayment strategy:

## 1. What's inside your `/tests` directory?

- A couple `ExampleTest.php` files and a whole lot of regret:  
**15 points**
- A handful of unit tests, but fewer than half the number of models in your app: **5 points**
- A handful of feature or integration tests, but fewer than half the number of controllers in your app: **5 points**
- A healthy assortment of unit, feature, and maybe even Dusk tests: **Nice job! 0 points**

2. Now, run your test suite. How many tests are skipped, failing, or contain errors?

- More than half: **10 points**
- More than a quarter: **5 points**
- None, all tests pass: **Congratulations! 0 points**
- Your test suite is automated to run in CI/CD:  
**score a -5 point bonus**

3. Does your codebase follow a consistent coding style, such as PSR-12?

- PSR-what? **10 points**
- Sometimes, but not always: **5 points**
- Religiously! **0 points**
- If you use an automated code style linter or fixer such as Tighten's Duster: **-5 point bonus!**

4. Search your codebase for comments you've left to your future self, such as `@todo`, `@fixme`, or `@refactor`.

- For each instance you find: **add 1 point**

## 5. Do your controllers need to go on a diet?

In your terminal, run:

```
wc -l app/Http/Controllers/*.php | sort
```

to view a list of your controllers sorted by their lines of code.

- For each controller over 300 lines: **add 1 point**
- For each controller over 1,000 lines: **add 5 points**
- For each controller over 2,000 lines (oh no!):  
**add 10 points**

## 6. Any “God objects” lurking amidst your models?

In your terminal, run:

```
wc -l app/Models/*.php | sort
```

to view a list of your models sorted by their lines of code.

- For each model over 250 lines: **add 5 points**
- For each model over 500 lines: **add 10 points**

## 7. Does your app have any potential “junk drawer” directories?

Directories such as **Services** or **Library** that contain files which aren't *really* services or libraries?

- If so, **add 5 points** for every file in there that you would be ashamed to show another developer. You know the ones.

## 8. It's time for a new developer to spin up a database on their local machine. Do they:

- Need to import a database dump: **10 points**
- Run migrations and seeders, but a couple of them might need fixing or updating first: **5 points**
- Run migrations and seeders without breaking a sweat: **0 points**

## 9. Scan your code for nested `if` and `foreach` statements.

For each one you find that is nested:

- 1 level deep: **add 1 point**
- 2 levels deep: **add 5 points**
- 3 or more levels deep: **add 10 points**

## 10. Time for some upgrades!

- My app is on the latest version of the Laravel framework:  
**Fantastic! -10 point bonus**
- My app is one major version behind: **add 5 points**
- My app is more than one major version behind: **add 10 points**
- For each dependency in your `composer.json` file that is no longer maintained or has been marked as deprecated:  
**add 5 points**

# Status Scale: Where Does Your Technical Debt Stack Up?

The previous questions are a good barometer to wrap your head around your technical debt situation. Now, use the scale below to find exactly what you should do next:

## 101+: Catastrophe



Quit your job, run away ugly-crying, move to Ecuador and raise alpacas (ok, we're kidding about quitting. But this *is* a technical debt emergency.)

## 30-49: Risky



Prioritize your next few tickets based on the areas of your codebase that scare you the most. Then, refactor a few things while you're tinkering around in there.

## 80-100: Emergency



Sharpen your debate skills — and get ready to fight your boss because you need to start paying down debt ASAP.

## 10-30: Stable



Make another coffee, then continue to push new features. You'll be ok... but start making a proactive plan for debt repayment.

## 50-79: Urgent



Start dropping hints that debt repayment should be a priority, like having your best dev make an expression of pure horror at the screen every time the boss walks by.

## 0-9: Excellent



Sigh with contentment and revel in a feeling of superiority.