

Attractive repository

Anton Grishin (@alchemmist)

EOSP #L6, winter 2026. CU x CPM



Table of contents

<code>intro</code>	A combination of factors.
<code>license</code>	How to choose a license.
<code>readme</code>	Recomendations for good readme.
<code>CI</code>	Automate all checks without installatino and configuring tools.
<code>make</code>	Entry point into repository. Useful tool.

Attractive = combination of factors intro

- Can I trust you? How do you position your code? `LICENSE.txt` explain that!
- What is it? How to use it? `README.md` or website explain that!
- How to configure it? How to up it locally? GitHub wiki or website explain that!
- Can I ask some questions? GitHub discussions explain that!
- How to report bugs? Issue templates explain that!
- How to contribute? `CONTRIBUTING.md` explain that!
- What are the rules in the codebase? CI explain that!
- Can I rely on this project? Speed of releasing explain that!
- And something else: time to merge, PL, stars, many contributors, many issues, ...

Success is predictable intro

«Predicting the Popularity of GitHub Repositories», 9th September 2016, Hudson Borges, Andre Hora, Marco Tulio ValenteASERG Group. doi: 10.1145/2972958.2972966.

- In study was developed some predication model.
- This model was try on a few of popular repositories for precation their growth in the next mounths.
- And model working not bad!
- For us it's mean, that if we will follow some rules, the probability of success will increase.

Table 3: Number of stars gained (real and predicted measures) for the top-10 (first table half) and bottom-10 repositories (second table half). Predictions are produced using a generic model ($t_r = 26$ weeks and $t = 52$). We can see that the error (column "% Diff") is lower for the top-repositories.

Repository	Stars	Predicted	% Diff
JQUERY/JQUERY	6,160	5,369	-12.84
ROBBYRUSSELL/OH-MY-ZSH	13,536	11,829	-12.61
AIRBNB/JAVASCRIPT	17,026	14,882	-12.59
H5BP/HTML5-BOILERPLATE	4,896	4,691	-4.19
METEOR/METEOR	9,919	10,082	+1.64
TORVALDS/LINUX	10,566	9,682	-8.37
DANEDEEN/ANIMATE.CSS	10,492	9,452	-9.91
FACEBOOK/REACT-NATIVE	18,443	19,373	+5.04
RAILS/RAILS	5,701	5,128	-10.05
DOCKER/DOCKER	10,268	9,721	-5.33
REACTIVERAVEN/JQBOOTSTRAPVALIDATION	213	298	+39.91
INFINITERED/PROMOTION	119	238	+100.00
NSLOCUM/DESIGN-PATTERNS-IN-RUBY	640	731	+14.22
JBT/MARKDOWN-EDITOR	621	744	+19.81
MUMBLE-VOIP/MUMBLE	565	667	+18.05
MANABU-GT/EXPANDABLETEXTVIEW	676	623	-7.84
APACHE/FLINK	890	712	-20.00
MAFINTOSH/MONGOJS	322	381	+18.32
ROFLOR/PROXYCHAINS-NG	813	790	-2.83
MIKEFLYNN/EGG.JS	584	793	+35.79

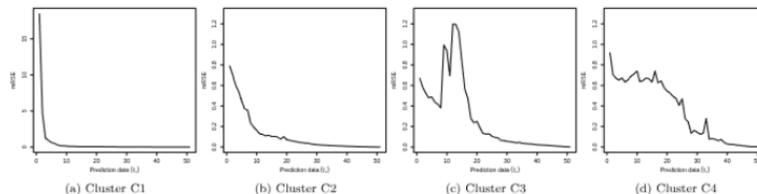


Figure 8: Model prediction error for different growth trends (i.e., clusters extracted using the KSC algorithm)

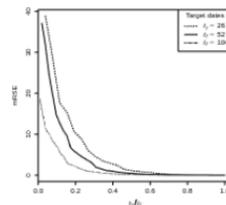


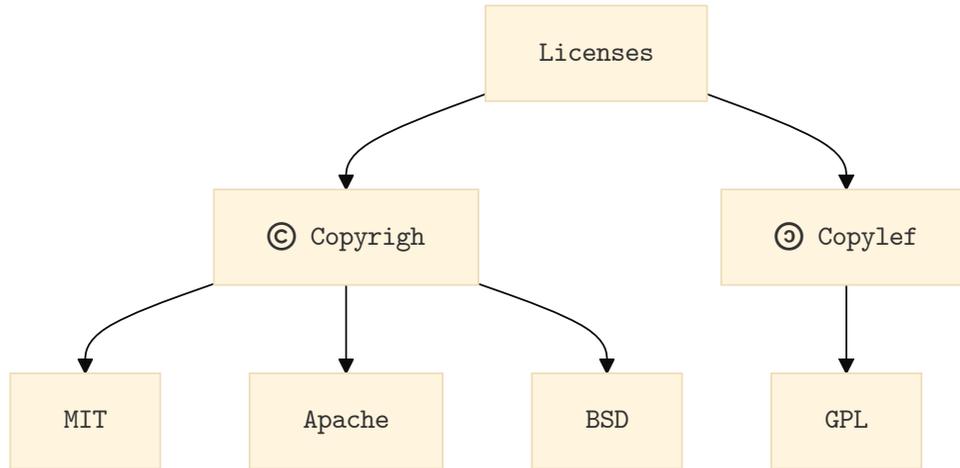
Figure 9: Generic model error (t_r -axis). Predictions for 26

we predict the number of stars at week 52.

Figure 8 reports the average error across all specific models. Cluster C1, which concentrates almost half of the repositories, presents a fast decreasing in the average error, e.g., $mRSE = 18.500 \pm 14.501$ for $t_r = 1$ week and 0.127 ± 0.020 for $t_r = 10$ weeks. This suggests that specific models for this cluster require very few data to provide accurate predictions. Cluster C2 also presents accurate results for any value of t_r . As we can observe, the accuracy of the models for cluster C2 is better than the accuracy for C1 when considering $t_r \leq 6$ weeks. However, for $t_r > 6$ weeks, the accuracy of C2 is slightly lower. For example, $mRSE = 0.030 \pm 0.009$ ($t_r = 26$ weeks) and $mRSE = 0.038 \pm 0.009$ ($t_r = 26$ weeks) for clusters C1 and C2, respectively. Cluster C3, which presents the

Your guarantee to users license

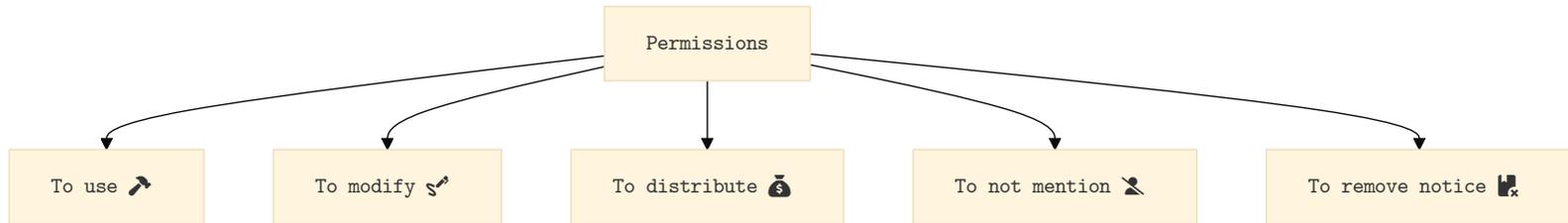
License tells your users and contributors how they are allowed to use your code.



The structure of license world license

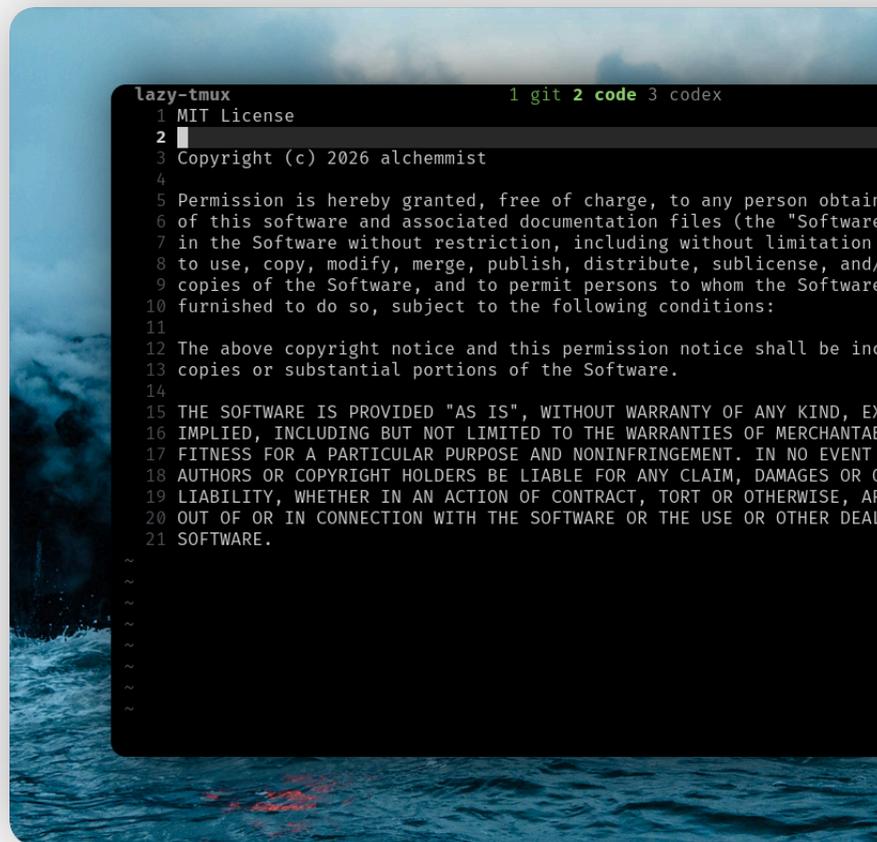
- Two categories of licenses: **copyright** and **copyleft**.
 - Copyright (permissive): allows almost any use of the code with minimal restrictions.
 - Copyleft: requires that derivative works keep the same license.
- Major licenses: **MIT**, **Apache**, **BSD**, **GPL**.
 - MIT: use, modify, distribute, sell; keep copyright and license text.
 - Apache 2.0: permissive + patent grant; patent rights terminate if you sue.
 - BSD: similar to MIT; no endorsement using the author's name.
 - GPL: strong copyleft; distributed modifications must also be GPL.
 - Overall it's a mess 😞.
- And many many more! You can even write your own license.
- For example: you could allow only students from France to use your code and prohibit everyone else.

Type of license premissions license



Recomendation: MIT license

- Very simple and permissive — almost anyone can use your code.
- Minimal legal overhead — just keep the copyright notice.
- Compatible with almost any other license.
- Encourages adoption — companies and developers feel safe using it.
- Flexible — allows closed-source derivatives if needed.
- Widely recognized and trusted in the Open Source community.



```
lazy-tmux 1 git 2 code 3 codex
1 MIT License
2 █
3 Copyright (c) 2026 alchemmist
4
5 Permission is hereby granted, free of charge, to any person obtain
6 of this software and associated documentation files (the "Software
7 in the Software without restriction, including without limitation
8 to use, copy, modify, merge, publish, distribute, sublicense, and
9 copies of the Software, and to permit persons to whom the Software
10 furnished to do so, subject to the following conditions:
11
12 The above copyright notice and this permission notice shall be inc
13 copies or substantial portions of the Software.
14
15 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EX
16 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTAB
17 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
18 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR C
19 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, AF
20 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEAI
21 SOFTWARE.
~
~
~
~
~
~
```

The face of repo `readme`

Let's see some examples. And define a few recommendations.

Is it good readme ?

The collage consists of four main panels:

- File Explorer:** Shows a directory structure with files like `src`, `tests`, `.env.example`, `.gitignore`, `.pre-commit-config.yaml`, `.python-version`, `Dockerfile`, `Makefile`, `README.md`, `docker-compose.yml`, `pyproject.toml`, `pytest.ini`, `requirements.txt`, `task.pdf`, `uv.lock`, and a highlighted `README` file.
- README Snippet:** Shows a section titled "Linux kernel" with a "Quick Start" section and a list of instructions: "Report a bug: See Documentation/a", "Get the latest kernel: <https://kernel.org/doc/Documentation/SubmittingPatches.rst>", "Build the kernel: See Documentation/Building", and "Join the community: <https://lore.kernel.org/>".
- GitHub Repository:** Shows the repository `glove80-keyboard` by `SylvanFranklin`. It lists files like `zmk-config`, `README.md`, and `glove80.kbd`. A snippet of the README shows "Horrible commands that don't work:" followed by a complex shell command: `;$s/$(2,)/ /g ; s/\v(\S+)\ze(\s+|$)/=print`.
- Course Completion Page:** Shows a course titled "Актуальные курсы" with a sub-section "Английский язык 105S1". A table shows the course status: "Статус: Сдано", "Вес: 20%", "Уровень: *** Без уровня", and "Оценка: 10/10". A button "Открыть задание" is visible.

And this one?

daktilo ("typewriter" in Turkish, pronounced "finger") is a small command-line program that offers the flexibility to customize keybindings and create an enjoyable typing experience.

✦ Inspiration: ["Taking notes in class"](#)

Now you can recreate this moment with **Warp**.

► Table of Contents

Getting Started

Simply run `daktilo` for the classic typewriter.

There are also different presets available:

Preset Name	Description
default	the classic typewriter
basic	an alternative and simpler
muchoboy	plays random notes

Resume (CV)

This is my Curriculum Vita (CV) as a software engineer. This repo has been rendered in `pdf` directory. This repo has whatever you want.

The `resume.cls` file is a class of document and visual view. I need to set it in a begin method.

```
\documentclass[margin, line]{resume}
\usepackage[utf8]{inputenc}
\usepackage[english, russian]{babel}
\usepackage[T1]{fontenc}
\usepackage{fontawesome}

\usepackage[absolute]{textpos}
\usepackage{enumitem}

\usepackage{graphicx, wrapfig}
\usepackage{url}
\usepackage[colorlinks=true, pdfstartcolor=blue, urlcolor=blue]{hypertex}
\pdfcompresslevel=9

\begin{document}
...
\end{document}
```

Warp

This is an issues-only repository for [Warp](#) where we discuss how developers work today, and agent-driven development.

We built Warp to solve two problems we know how developers work today, and agent-driven development.

Warp is a modern terminal built for coding, modern UI and code editing features. Use Code, Codex, or Gemini CLI.

PDM

A modern Python package and dependency manager supporting the latest PEP standards.

DOCS MKDOCS GET UPDATES DISCORD 25 ONLINE

Tests failing pypi v2.26.6 codecov 86% Packaging status downloads/week 572k

```
21:29:35 #fainq@FROSTING-M80 ~/workspace - system
$ pipx install test-pdm
21:29:39 #fainq@FROSTING-M80 ~/workspace - system
$ cd test-pdm
21:29:41 #fainq@FROSTING-M80 ~/workspace/test-pdm - system
$ pdm init
Creating a pyproject.toml for PDM ...
Please enter the Python interpreter to use: 3.8
./Users/fainq/Library/Python/cwd/python3.8 (3.8.5)
1 ./usr/bin/python3 (3.8.2)
Please select: [0]:
Using Python interpreter: ./Users/fainq/Library/Python/cwd/python3.8 (3.8.5)
Project name [test-pdm]:
Project version [0.0.0]:
License (SPDX name) [MIT]:
```

Recommendations readme

- ...