QCon
全球软件开发大会

中小团队的前端工程化实践

白鹭引擎 首席架构师 / 王泽

极客时间 | 企业版

极客时间
SVIP团队体验卡

畅学千门IT开发实战课

「扫码免费领课」

为什么选择这个题目

客户端工程化现状

前端工程化“行业”现状

旱的旱死涝的涝死

面向观众

中小型团队研发骨干

什么是“工程化”

一切以提高效率、降低成本、质量保证为
目的的手段，都属于工程化

工程化概述 - 知乎 (zhihu.com)

前端工程化的一些原则

- 使用单一技术平台，而非整合一些更好的技术平台（们）
  - 正面例子：GitLab + GitLab CI+ Gitlab Issue + Gitlab Wiki
  - 反面例子：GitLab + Jenkins + JIRA + Other Wiki + LDAP

前端工程化的一些原则

- 遵循技术平台的限制，而不是尝试定制符合自己公司需求的工作流
- 对一家中小型企业来说，如果贵公司的工作流和主流技术平台的工作流相违背，90%的原因是贵公司的工作流有问题，而不是主流技术平台的工作流有问题。

- 典型案例：
  - “领导要求把缺陷追踪软件的【被分配人】设置成多个人“
  - “错的不是我，是这个世界”

前端工程化的一些原则

- 工程化是用来解决问题的，而不是制造麻烦的。“有多大锅下多少米”。
- 典型案例：
  - 绝大部分缺陷追踪软件中的时间统计功能对中小型团队而言毫无意义。

工程化两大关键点

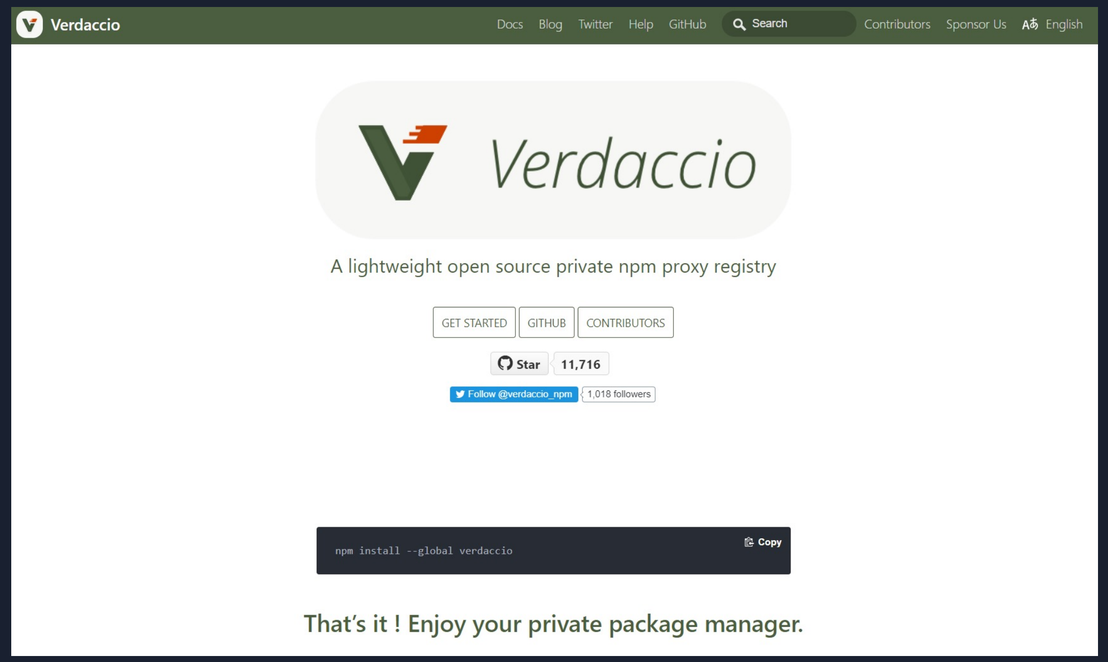
- 尽可能的代码复用
- 尽可能的自动化测试

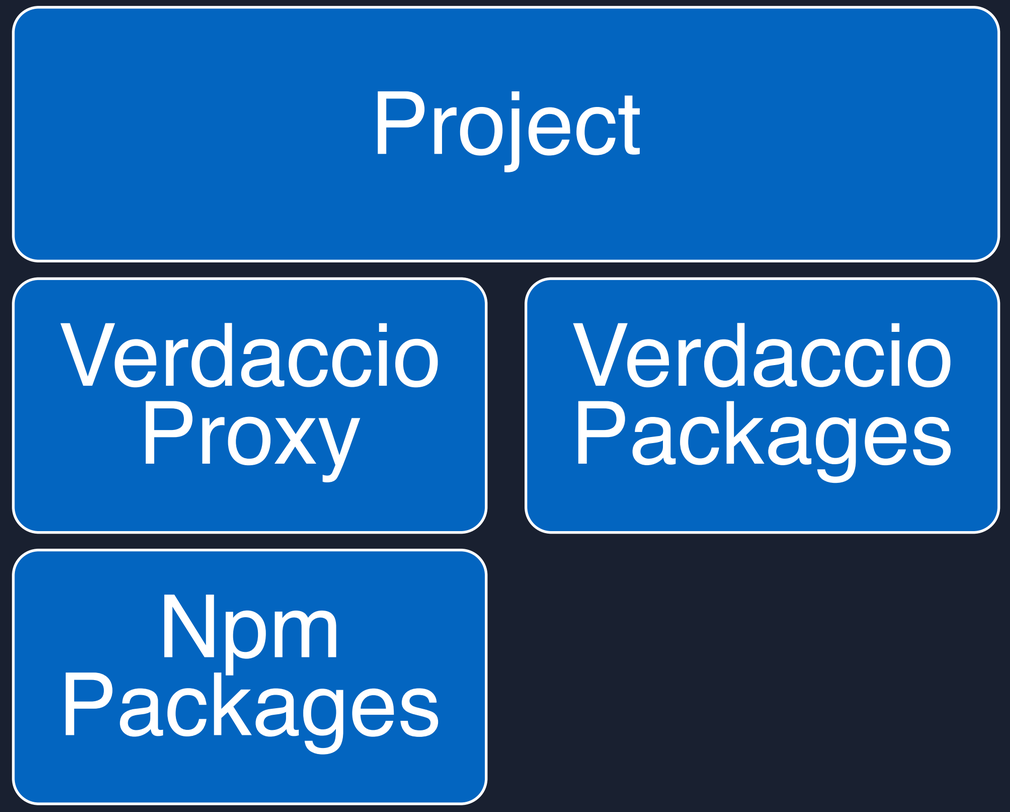
如何更好的实现复用

如何更好的实现复用

如何更好的实现复用

- 如何优雅的调用自研库

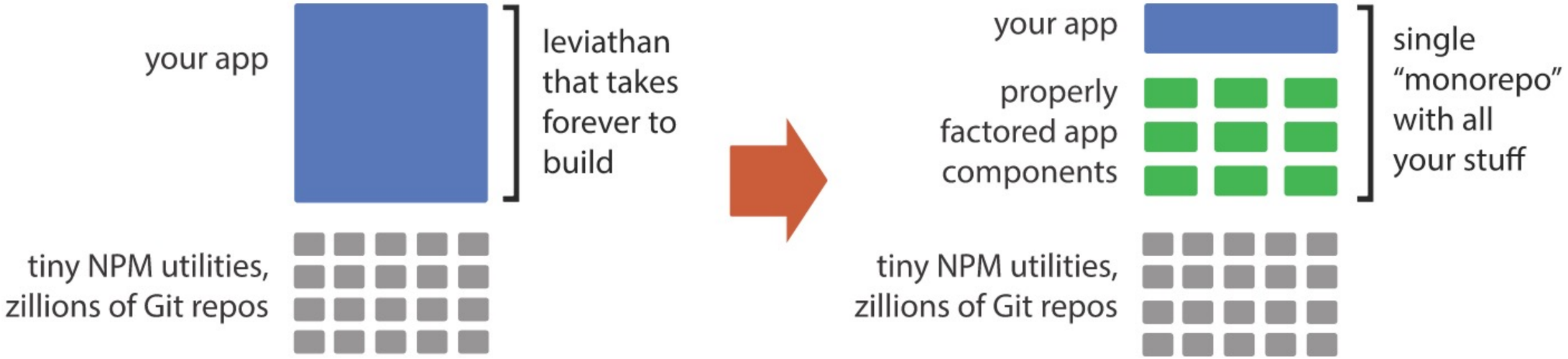




如何维护自研库

如何维护自研库

使用 Monorepo


your app
leviathan
that takes
forever to
build
tiny NPM utilities,
zillions of Git repos
your app
properly
factored app
components
single
“monorepo”
with all
your stuff
tiny NPM utilities,
zillions of Git repos

备受争议的Monorepo

Monorepo 是什么，为什么大家都在用?

刘留
http://liuliu.me

67 人赞同了该文章

Dan Luu 很早很早就写了篇文章，给大家介绍 monorepo 。在我之前那篇推荐 Buck / Bazel 的文章之后就想讲讲 monorepo，结果一直没来得及写。

Monorepo 的概念要和互联网公司里怎样训练新人上手一起讲。很多公司要花超过半个月的时间才能让新人开始动手干活，并不是内部系统要学的东西很多，只是因为有很多的部落知识（Tribe Knowledge）并没有写下来，需要手把手地教。其中一项，就是怎么把项目的所有依赖从网上拉下来，配置好开发环境，可以干活。之后，还要教为什么提交的修改不应该包含一些配置修改之类的，或者为什么要用祖传数年前的依赖版本。

这些如果用 monorepo 的话都不是问题。

Monorepo 简单的说，是指将公司的所有代码放到一个 Git / Mercurial / Subversion 的代码仓库中。对于很多没听说过这个概念的人而言，无异于天方夜谭。Git 仓库不应该是每个项目一个吗？对于很多用 monorepo 的公司，他们的 Git 仓库中不止有自己的代码，还包括了很多的依赖。基本上，只要把 monorepo 用 Git 拖下来，跑一下 `./scripts/install`，就可以直接用 Buck / Bazel （在安装脚本中就装到了本地）编译仓库中的所有项目，并且提交修改（安装脚本配置好了代码提交环境，如果用的 Phabricator 的话，Gerrit 不用）。

备受争议的Monorepo

羲和 2020-01-15

感觉是个花哨的概念

4 回复 踩 举报

知乎用户 04-02

缺点就是某个周五的下午一个公司新人提交了一段神秘代码，然后周末整个公司的项目全都挂了

3

思无邪 2019-08-24

b站好像就是这个思路，然后代码被泄露了

2

Hessian 回复 思无邪 2019-09-10

B站只是后端部分吧

赞

思无邪 回复 Hessian 2019-09-10

我大概浏览了一下泄漏的那个代码，感觉不止是后端部分，好像是除了前端之外的所有东西

赞

查看全部 10 条回复

邵锁 02-26

需要也确实需要，多个仓库本地修改立即生效，不得不用monorepo。灾难的话还是在于如何控制版本不一致的问题。 ABCDE项目版本完全匹配，项目才能跑起来。不过这个可以用CI检测build 的问题

1

莱恩叔叔 02-25

没明白解决了什么问题

1

xuwanlin 02-22

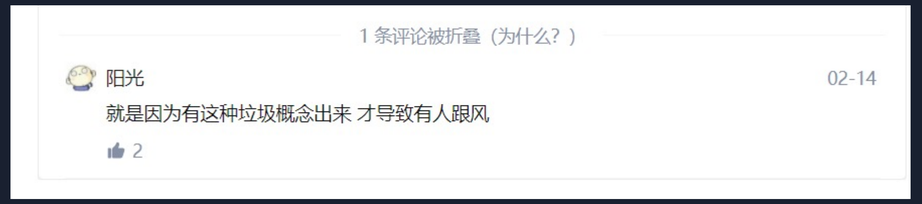
不明白你在说啥

1

阳光 02-14

还自以为很先进

1



1 条评论被折叠（为什么？）

阳光 02-14

就是因为有这种垃圾概念出来 才导致有人跟风

2

备受争议的Monorepo

- 一处代码泄露导致全公司所有项目裸奔。
- 一次提交导致构建执行时间超长。
- 一行代码导致全公司所有项目挂掉。
- 配置过于繁琐。
- 学不动了。

- 业务代码与框架代码分离
- 增量构建
- 编写自动化测试，规划版本配置。
- 使用开箱即用的 rush
- ？？？？

Why Rush

- 其他的解决方案
  - Lerna
  - Yarn workspace
  - Npm workspace
- Rush 的优势：不仅是 monorepo，更是版本配置管理工具。
- Rush 的劣势：在一些场景下比较“固执”

Rush，开箱即用的Monorepo解决方案

- 扁平化依赖安装，降低 node_modules 尺寸
- 增量构建与构建缓存机制，解决构建速度问题
- 规范化 CHANGELOG 文件编写
- 规范化版本发布与版本号修改
- 一键部署
- 自定义命令扩展

如何更好的实现复用

工程化两大关键点

- 尽可能的代码复用
- 尽可能的自动化测试

自动化测试是否对中小团队过于奢侈

- 自动化测试包括哪些内容?
  - 集成测试
  - 模块测试
  - 单元测试
  - 静态代码检查

不积跬步无以至千里

- 从静态代码检查做起
  - 统一的代码格式化工具与配置　prettier
  - 统一的 Lint 规则　ESLint
  - 自定义检查工具

勿以善小而不为

- 自定义检查工具
  - 拼写检查工具　https://www.npmjs.com/package/cspell
  - 配置文件检查
  - 基于抽象语法树的代码检查

检查时机

- Git Hooks 还是 Gitlab Pipeline
- 个人推荐
  - 使用标准 Git 工作流，分支开发 + Merge Request (Pull Request) 的方式提交代码
  - 在 Merge Request 时通过 Gitlab Pipeline 进行自动化工作，Pipeline 执行失败的化不允许执行 Merge 操作
  - 如何避免 Merge Request 的 Code Review 变成走过场
- 个人建议中小型团队不要使用 Git Hooks

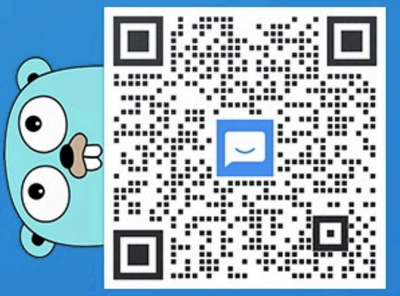
总结

- 千里之行，始于足下
- 优先做投入产出比高的事情
  - 自研库
  - CodeReview 与静态语法检查

THANKS

抢占先机，成为未来 3 年抢手的后端开发人才

【Go 进阶训练营】带你成为字节跳动 2-2 级别 Go 工程师



扫码获取详细大纲
并咨询课程详情

- 12 大模块教学
- 3 大企业级实战案例
- 13 周视频课程教学
- 大厂助教 1v1 答疑
- 班主任全程贴心督学
- 简历直推一线互联网公司