前端搞基建

1. 前言

偶然一次机会参加了早早聊大会【基础设施】主题分享，其中堂主‘前端如何搞基建’主题分享内容非常棒，所以我经过吸收整理之后拿出来给跟大家分享，本次分享是面向前端同学的，但是这种思路后端同学也可以借鉴一下

，

二、前端常见吐槽

同学一：xxx项目太烂了，天天不是在修复bug就是在写bug，感觉这样下去要废了。。。

同学二：天天 写业务代码，感觉没什么技术含量，也没什么沉淀，成长太慢了，好想做技术架构啊，但是又不知道如何下手。。。

同学三：公司最近业务太繁忙了，每天不是在加班就是在加班的路上，身体吃不消啊。。。

总结上面的同学的吐槽都是因为研发效能低和个人得不到成长而感到身心疲惫、前途渺茫！我们面对这些困境/瓶颈我们要如何破局呢？这里我要引出今天的主题：前端搞基建。

三、理解基建

1、什么是基建

造轮子能力

2、为什么要搞基建

成本控制：

对于一个研发团队，如果一直都是靠压榨、纯加班这种出蛮力的方式在支持业务，这个团队会非常危险，业务也会危险。

出蛮力模式下，业务是无法实现跨越式增长的 —— 你总不能指望业务量增长 10 倍的情况下研发团队规模也扩充 10 倍，成本会失控。

有时候阶段性的忙和加班是不可避免的，比如电商的双 11 大促，或者 toB 业务定制的大项目的交付，时间点都是倒排，守时履约对结果的影响非常重。加班是应该的，不加班也是应该的，只有完不成工作是不应该的。当这一阵过去后，团队一定要思考，怎么做能更高效？

站在未来看今天，如果一年、两年后，业务量增长 N 倍，那时候该如何支持，现在的方式是否能满足今后？不可能靠堆人，只能靠技术建设去提效降成本。这就是基建最核心的价值：帮助业务更好的活在未来

 3.基建的意义

* 解决问题：帮助解决业务问题。
* 团队练兵：承担建设的虚拟团队，在建设过程中能为同学提供不同维度的锻炼场景，在业务问题与场景的识别、方案设计、新技术实践、项目管理和产品化思维方面都能提供实践成长的空间，起到练兵的作用。
* 梯队建设：一个虚拟建设小组本质上也是一个团队，过程中能对不同角色进行锻炼和考察，这有助于团队梯队的完善。
* 影响力建设：建设结果对于业务的促进，更容易获得内部合作方的认可；沉淀下来的好的经验，可以对外输出分享，也是对影响力的有力帮助

五、基建怎么搞

思路

1.开发整个链路全面复盘 -》找出堵塞点=〉解决问题

一个基本的研发流程闭环，一般是需求导入 **=>** 需求拆解**=>** 技术方案制定 **=>** 本地编码 **=>** 联调 **=>** 自测优化 **=>** 提测修复 Bug **=>** 打包 **=>** 部署 **=>** 数据收集&分析复盘 **=>** 迭代优化 —— 即新一轮的需求导入。

在这个基础的闭环中，每一个节点都有其进一步的内部环节，每一个环节相连，组成了一个研发周期。这个周期顺，研发流程就顺。这个周期中每一个环节的阻塞点越少，研发效率就越高。最初期的基建，就是从这些耽误研发时间的阻塞点入手，按照普遍性 + 高频的优先级标准，挨个突破，最后实现标准化 + 规范化 + 工具化 + 自动化+平台化+产品化

2.具体case

1.规范文档

规范意味着标准，是团队的共识，是沟通协作的基础。其中包括编程规范，git规范。。。，标准不一定要定的多么华丽，团队认同就行。

2.本地工程化环境cli

每个团队都会有自己的本地环境cli标配，可能就是一个vue-cli，react-cli，或者自己用webpack写一个

脚手架，其目的都是为了本地开发流程的去人肉化、自动化。

具体怎么做呢？

比如可以在**cli**界面通过输入**pro run** 的方式，实现了项目初始化到打开浏览器的动作！省了 npm install 和 dev 的步骤。

另外还可以ui组件、mock、埋点做成插件的方式安装到项目，Cli 的意义就是抹平选型的差异，降低入手成本，一键化解决所有问题

3.可视化工程系统gui

比cli更体现人效的方式是 gui，大家看一下这个系统，这是一个桌面客户端工具，里面几乎囊括了前端所有相关操作，他将日常的前端研发链路上的操作都聚合进来，从组件开发到模板开发再到应用开发；从唤起编辑器到启动调试环境、进行包更新到打包部署发布。这可以说是基建的终极目标了

1. 项目工程管理

项目在创建、本地环境搭建、分支管理，构建部署前端一个人搞定，不需要因为权限问题找人帮忙建仓库，也不要组件、区块/模板选型等不一致而学新，不需要人肉配脚本打包，也不需要每次部署都要找运维啊等等，总之工具帮我们省了很多的时间让我们更专注编码

1. 组件开发与管理

通过工具对组件进行创建管理，可以更好的实现命名标准化，开发流程的简化，也可以统计组件的使用率，让我们更好的定向优化

6.模版开发与管理

管理后台经常用的表单，表格经常标准作出一套模板，根据场景选择合适的模板，配置信息和路径后一键安装到本地并配置好路由和依赖

7.前端基础资产

8.CICD 自动化构建部署

目前前端小组已经具备了半自动的方式部署，可以通过手动执行脚本进行前端打包部署，自动化的姿势是执行git push提交代码即可进行lint检查，兼容API检测，https检查，黑名单，404等检测最终发布

1. 可视化搭建系统

可视化搭建系统是进一步高效利用组件的上层建筑，就是由人肉对组件进行拼装，从界面搭建到构建部署，数据埋点都可以进行配置，赋予了产品、运营更多的能力。比如像一下落地页就可以通过搭建这样的系统给运营同学用，解放前端生产力，可视化搭建系统跟上面的可视化工程系统他们服务的对象是不一样，前者是面向前端，后者面向运营和产品。

1. 数据埋点与分析

11.自动化数据分析

从立项到上线运营全链路复盘

Cli 的意义就是抹平选型的差异，降低入手成本，一键化解决所有问题，组件，插件化（mock，买点）集成

六、因为你有什么变得不一样