

# 管理视角看效率：转转研发效能管理

李金城  
转转-技术VP

# 目录

- 技术基建对效率的影响
- 技术选型对效率的影响
- 历史包袱对效率额影响
- 组织对效率的影响
- 日常效率管理保障体系

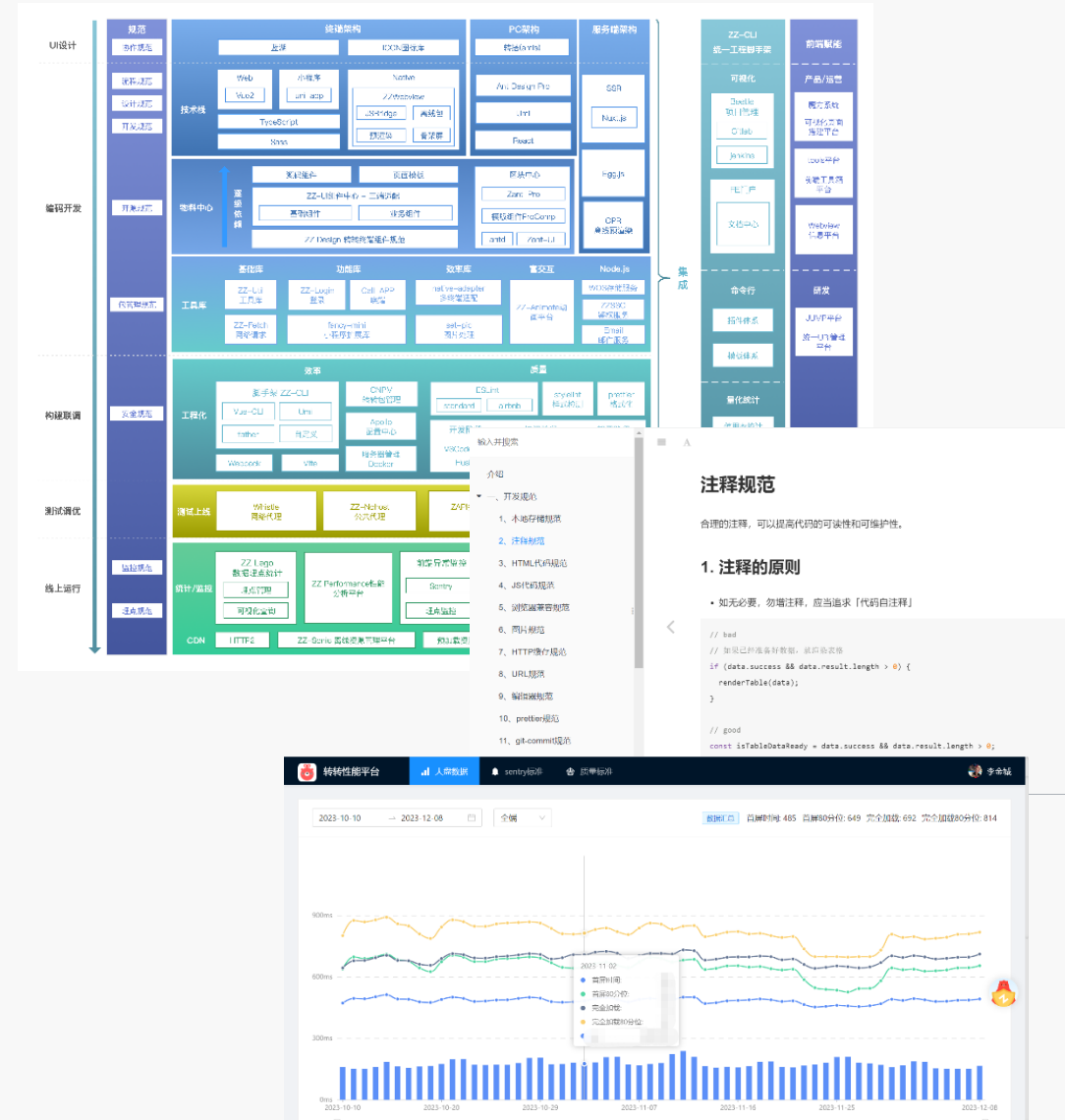
# 统一是基建最快的路（技术基建对效率的影响）

2015~2018 转转FE没统一技术栈，各种新技术，重复造轮子，但基建却几乎没有，随着团队扩大，效率问题、质量问题频发，18年开始加强基建建设。

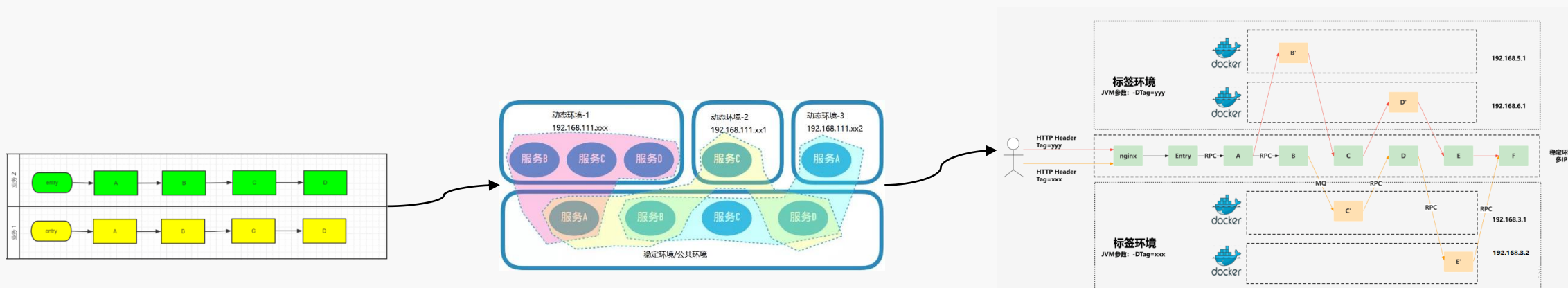
- 方向确定后，成立前端技术委员会，一起决策，**最大化共识**
- 从业务调人，扩大前端基础架构团队，**专人专事**持续建设
- 从统一开发框架开始，前台VUE，后台React，**聚焦沉淀**
- 修改**晋升标准**，逐步加强统一基建共识，严控重复造轮子

19年底局面开始扭转，随着基建日趋完善，团队从忙于救火转为追求卓越。

- 组件库组件占比 **67%**，页面开发效率提升 **30%**，UI还原度 **90%+**
- 运营自助生成运营活动占比 **95%**
- 页面平均打开时间低于 **0.8s**



# 高频使用的基建，不能只是够用（技术基建对效率的影响）



## 转转测试环境治理过程：

- 2017年成立之初，转转测试环境只有5台64G内存的老机器，搭5个完整的测试环境，够用，但不太好用。
- 2019年上线了基于IP路由的动态增量部署方式，硬件资源下降30%，够用，比以前好一些。
- 2021年上线了基于标签路由的动态增量部署方式。

稳定性提升 **95%**

部署时间5分钟以内，降低 **70%**

资源自动回收，从消耗3200G到1200G，降低 **60%**

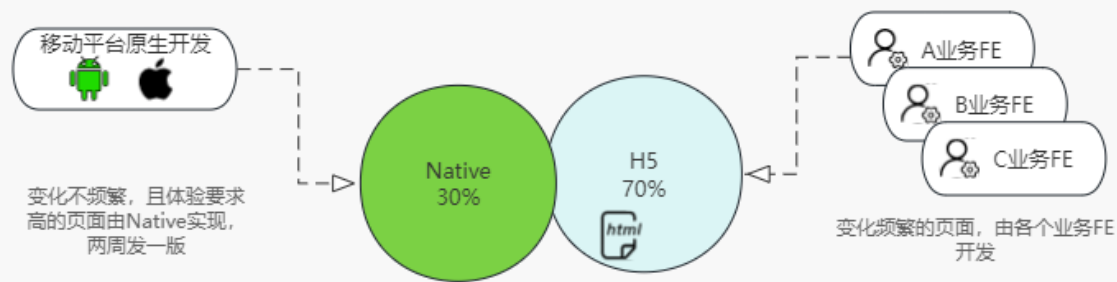
回想起来，理想方案应该更早研发。所以高频使用的基建不能只是够用，要尽早做到位，复利带来的收益远超一时的投入。

## 技术方案选对了，事半功倍（技术选型对效率的影响）

2016年，转转APP基于Native开发，一月一版业务觉得慢。

开始试着加大客户端投入，两版并行两周一版，连轴转，问题有好转但人受不了。

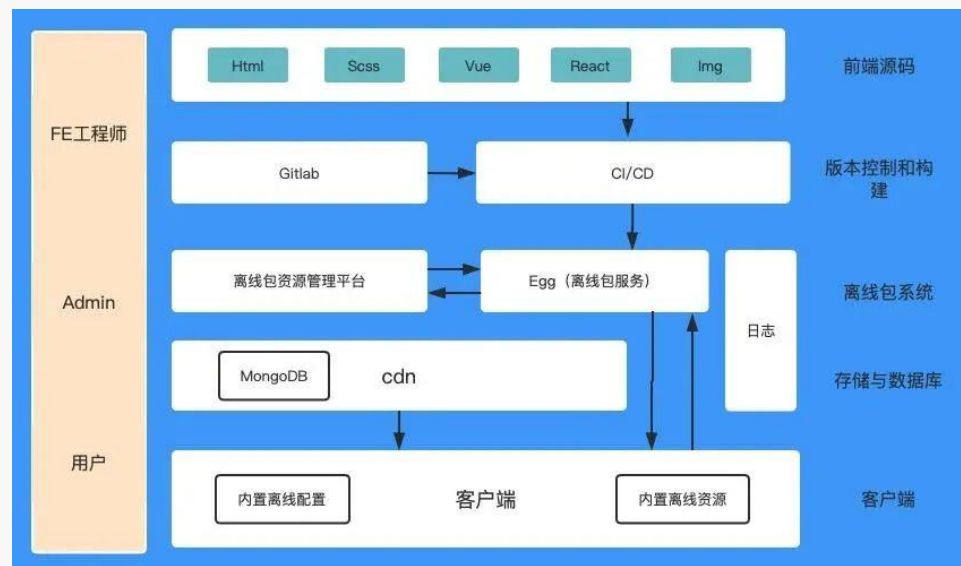
最终转为Hybrid开发模式，70%的页面交给FE开发，发版快还省人，问题轻松解决了。



- 2016年之后再没出现过发版瓶颈问题
- H5比双端原生开发效率提升了**40%**

当然也引入了体验不如Native的问题，我们通过预渲染技术，H5的体验问题大幅改善。

- 高PV页面首屏时间: 485ms
- 高PV页面加载完成: 693ms



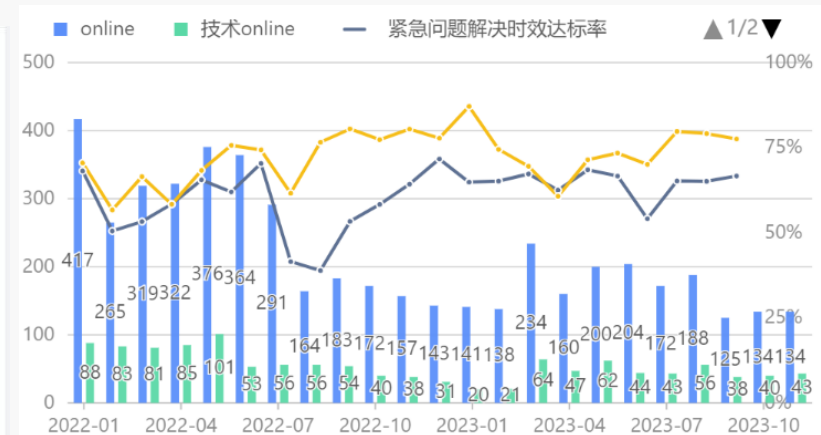
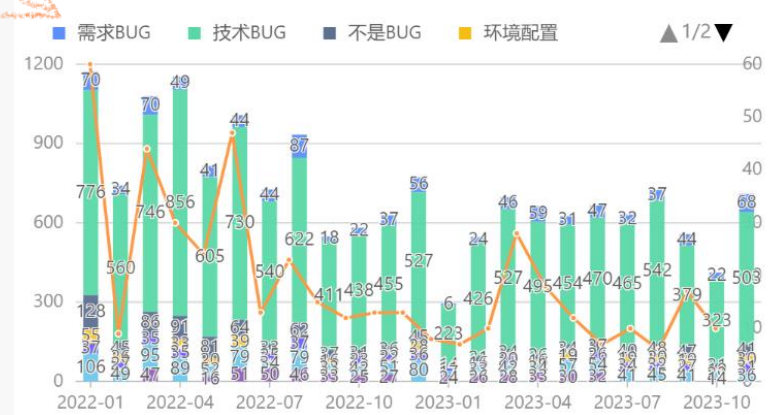
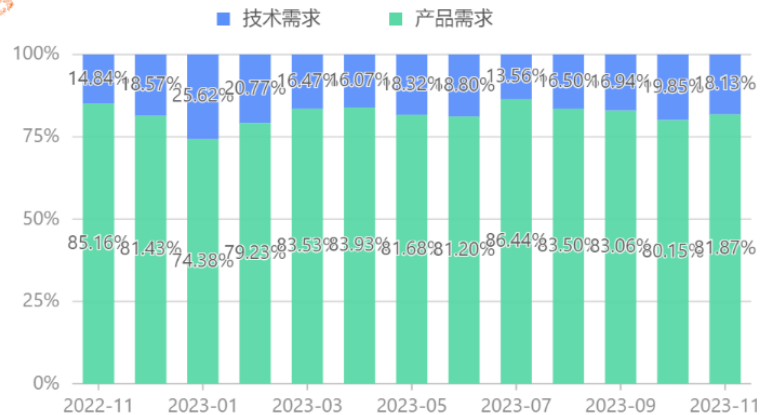
技术不能封闭，要走出去，加强交流，扩大视野。21年转转联合去哪儿创建ITCP联盟，效果不错。

# 控制技术债（历史包袱对效率的影响）

技术债重要不紧急，常常被忽视，积少成多，慢慢会导致效率和稳定性下滑，如果拖到问题集中爆发，那影响很差，解决成本非常高。

防微杜渐的一些方法：

- 大债要记录，强制保持10%~20%技术建设投入。年底1-2月业务规划阶段，集中瘦身、还债。
- online问题跟进机制，监控线下和线上质量趋势。







# 测试团队要尽量小（组织对效率的影响）

2017年，转转QA主要负责功能测试，业务增长很快，QA人口增长更快，人多了质量还在降低。



要从“保姆”向质量保障转型。

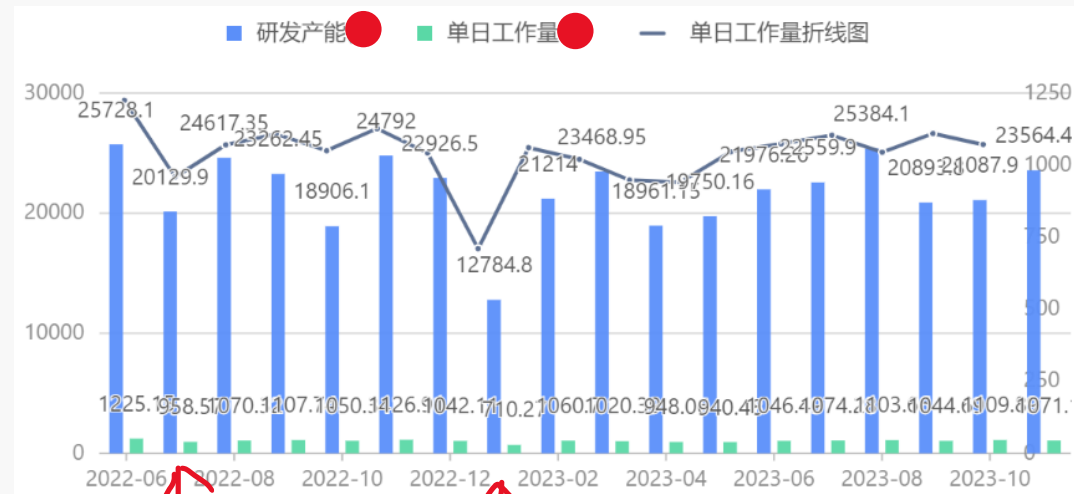
- 聚焦核心项目：强制推行开发自测率50%，边缘系统QA不介入，QA工作减少30%，质量基本没影响。
- 提高提测质量：控需求优先级、抓需求质量、编码质量、冒烟case通过率等，bug/case持续降低，后面发现的问题少了，效率自然提升了。
- 扩大测试开发比例，加强测试平台建设。例如：case管理、接口管理、自动化测试能力、数据构造能力、自动化压测能力、兼容性测试能力等等，大幅降低了测试成本。

开发测试人数比从17年 2:1 到了 **4:1**。

开发测试周期比从17年 1.2:1 到了 **2:1**。



日期	编码时间 (%)	其他 (%)
2022-06	49.3%	50.7%
2022-07	47.7%	52.3%
2022-08	50.9%	49.1%
2022-09	51.9%	48.1%
2022-10	52.7%	47.3%
2022-11	54.8%	45.2%
2022-12	54.0%	46.0%
2023-01	45.4%	54.6%
2023-02	56.6%	43.4%
2023-03	57.3%	42.7%
2023-04	54.5%	45.5%
2023-05	54.2%	45.8%
2023-06	55.6%	44.4%
2023-07	56.7%	43.3%
2023-08	57.2%	42.8%
2023-09	55.0%	45.0%
2023-10	56.8%	43.2%
2023-11	55.1%	44.9%



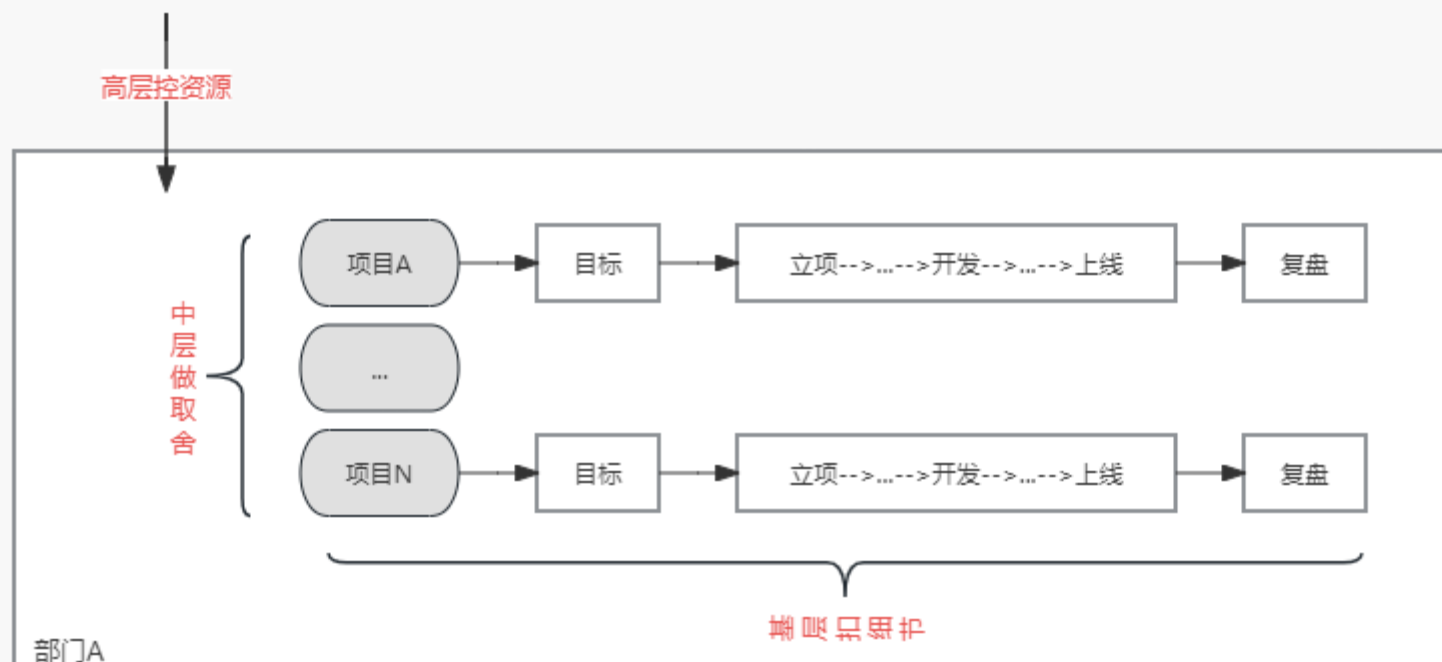
- 紧抓资源：人少效率高，人少思考多。
- 保持流动性：每年保持10%~20%人员流动，一定只补优秀的
- 解决Leader沟通拖延问题：双月绩效沟通机制，制定沟通模板表格，每双月底HR提醒Leader进行沟通，有效解决沟通拖延的问题。

2023年度项目：5G网络优化项目		项目信息		项目计划									
阶段	开始时间	结束时间	负责人	状态	备注	计划开始时间	计划结束时间	计划负责人	计划备注	计划开始时间	计划结束时间	计划负责人	计划备注
1	2023/1/15	2023/1/31	张三	进行中	项目启动，完成需求调研。	2023/1/15	2023/1/31	张三	项目启动，完成需求调研。	2023/1/15	2023/1/31	张三	项目启动，完成需求调研。
2	2023/2/1	2023/2/28	李四	已完成	完成需求调研，制定项目计划。	2023/2/1	2023/2/28	李四	完成需求调研，制定项目计划。	2023/2/1	2023/2/28	李四	完成需求调研，制定项目计划。
3	2023/3/1	2023/3/31	王五	进行中	完成需求调研，制定项目计划。	2023/3/1	2023/3/31	王五	完成需求调研，制定项目计划。	2023/3/1	2023/3/31	王五	完成需求调研，制定项目计划。
4	2023/4/1	2023/4/30	赵六	进行中	完成需求调研，制定项目计划。	2023/4/1	2023/4/30	赵六	完成需求调研，制定项目计划。	2023/4/1	2023/4/30	赵六	完成需求调研，制定项目计划。
5	2023/5/1	2023/5/31	孙七	进行中	完成需求调研，制定项目计划。	2023/5/1	2023/5/31	孙七	完成需求调研，制定项目计划。	2023/5/1	2023/5/31	孙七	完成需求调研，制定项目计划。
6	2023/6/1	2023/6/30	周八	进行中	完成需求调研，制定项目计划。	2023/6/1	2023/6/30	周八	完成需求调研，制定项目计划。	2023/6/1	2023/6/30	周八	完成需求调研，制定项目计划。
7	2023/7/1	2023/7/31	吴九	进行中	完成需求调研，制定项目计划。	2023/7/1	2023/7/31	吴九	完成需求调研，制定项目计划。	2023/7/1	2023/7/31	吴九	完成需求调研，制定项目计划。
8	2023/8/1	2023/8/31	郑十	进行中	完成需求调研，制定项目计划。	2023/8/1	2023/8/31	郑十	完成需求调研，制定项目计划。	2023/8/1	2023/8/31	郑十	完成需求调研，制定项目计划。
9	2023/9/1	2023/9/30	冯十一	进行中	完成需求调研，制定项目计划。	2023/9/1	2023/9/30	冯十一	完成需求调研，制定项目计划。	2023/9/1	2023/9/30	冯十一	完成需求调研，制定项目计划。
10	2023/10/1	2023/10/31	陈十二	进行中	完成需求调研，制定项目计划。	2023/10/1	2023/10/31	陈十二	完成需求调研，制定项目计划。	2023/10/1	2023/10/31	陈十二	完成需求调研，制定项目计划。
11	2023/11/1	2023/11/30	林十三	进行中	完成需求调研，制定项目计划。	2023/11/1	2023/11/30	林十三	完成需求调研，制定项目计划。	2023/11/1	2023/11/30	林十三	完成需求调研，制定项目计划。
12	2023/12/1	2023/12/31	周十四	进行中	完成需求调研，制定项目计划。	2023/12/1	2023/12/31	周十四	完成需求调研，制定项目计划。	2023/12/1	2023/12/31	周十四	完成需求调研，制定项目计划。

# 日常效率管理保障体系（思路）

日常效率管理的三件事：

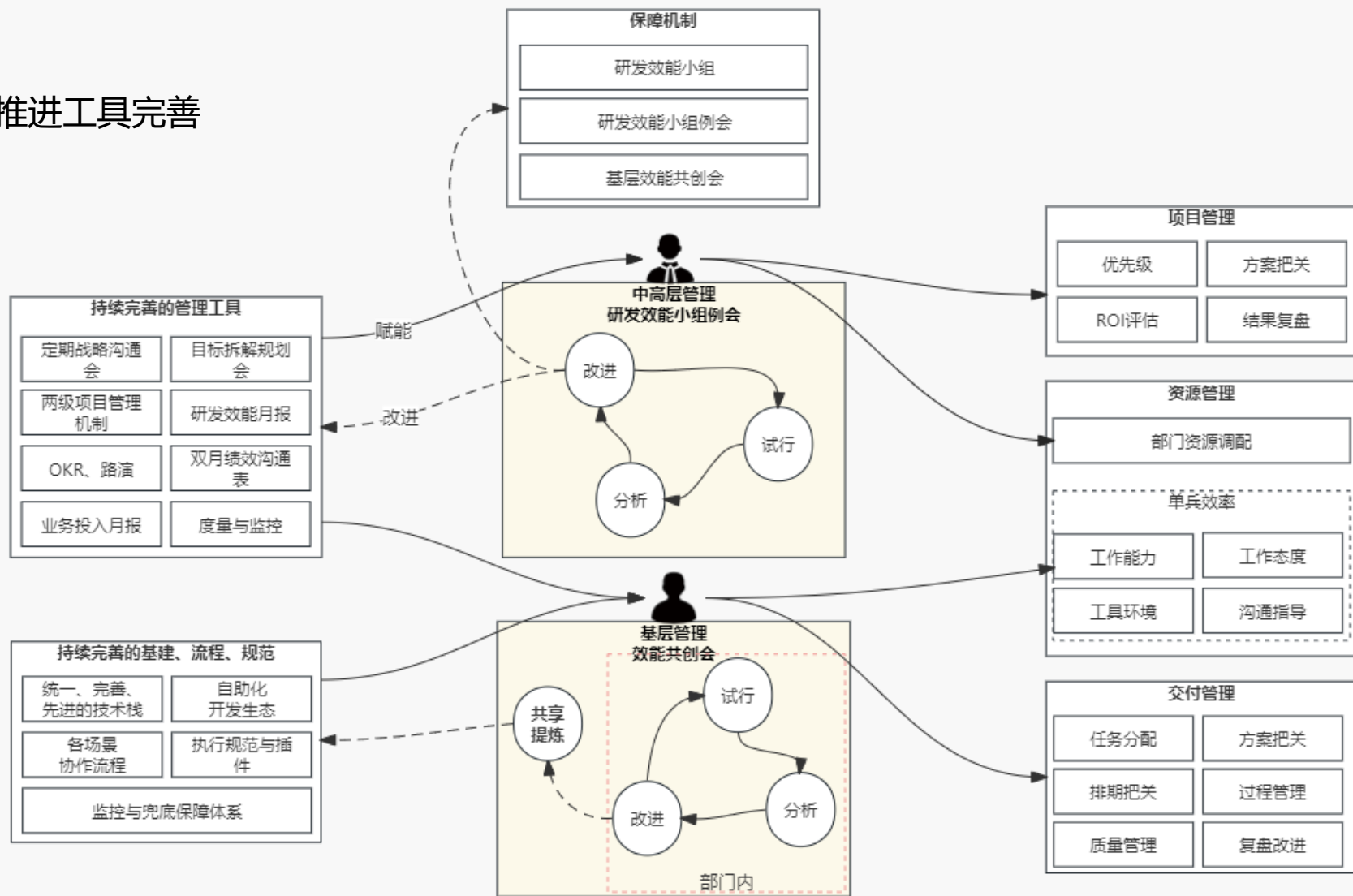
- 项目管理：分辨并聚焦于重要的事情
- 资源管理：合理分配资源（HC），不断提高单兵投入度
- 交付管理：不断降低做事成本



如何能持续迭代？

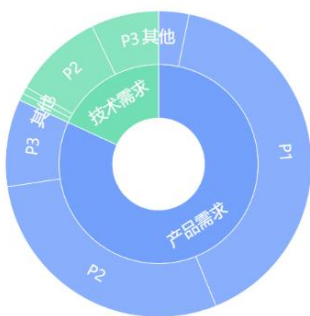
# 日常效率管理保障体系（保障机制）

工具赋能人来管理，机制推进工具完善



# 日常效率管理保障体系（推荐三个效率指标）

需求价值分布

[?](#) [导出](#) [全屏](#)


■ 产品需求 ■ 技术需求

## 需求价值分布

产品需求和技术建设的投入比例。产品需求各个优先级分布。

这个指标能反映出资源都投在哪，比例合不合理。

负载率

[?](#) [导出](#) [全屏](#)


## 研发负载率

开发花在编码工作上的时间与工作时间的比。

这个指标能反应出单兵的工作效率。

产品需求响应率

[?](#) [导出](#) [全屏](#)


## 需求响应率

统计周期内已确认的产品需求进入研发环节的占比。

这个指标结合前面两个指标，能反映出资源多了还是少了。

任正非说

“我们一定要避免封闭系统。我们一定要建立一个开放的体系，特别是硬件体系更要开放，不开放就是死亡。”





金城

广东 深圳



扫一扫上面的二维码图案，加我为朋友。