每个人都在自己本地静态代码扫描也结束了，单元测试覆盖率报告也都出来了

每个人提交Pull Request给其他同事，或者是leader，来review这个代码，leader review代码的时候，主要看4样东西：

1、他会去看每个人的每一行代码，3个目的

作为一个leader，或者是团队里其他的高工，你有责任和义务去仔细看其他人的代码，如果出问题，你作为leader或者高工有责任去帮低级别的人解决问题。

code review，就是一个完美的时机，你作为一个leader或者高工，去仔细看手下写好的代码，你这样才可以把控全局，控制所有的东西。别让自己掉队。万一你手下小弟刚写好代码，外面拿到个offer，薪资暴涨50%，人跑了。

code review，你或者其他高工看懂别人的代码，就代表你成为了那个人的一个backup。如果那个人跑了，你可以直接顶上去。

1）帮低级别的人解决问题

2）backup

3）掌控系统的全局，每一块代码大概是怎么写的，心里有数

2、你作为一个检查机制，检查在code review之前的规范，有没有执行

别人给你给你提交PR的时候，你必须得让他同步提交三样东西：

1. 静态代码扫描的结果：blocker和critical级别的问题是否都解决了，major问题遗留了多少个
2. 单元测试覆盖率的报告：你去检查他负责的那几个模块，单元测试的覆盖率是否达标
3. 冒烟测试报告：他对他负责的模块的核心流程都进行了冒烟测试，确保他负责的模块的核心流程都可以跑通

你要作为一个leader，去检查他都做了这几件事情

1. 以你的技术能力和经验，去发别人代码中的不合理之处，指导低级别的工程师去修缮和完善自己的代码

优秀的code review，code review的时候，要注意哪些事情，1,2,3,4,5,

我曾经有一段时间再我的团队里也推行了类似的这种code review的机制，我写了好几千字的code review规范。谁能记住？

连我自己都记不住。谁会在review代码的时候，一边review代码，一边对照着规范来检查。傻乎乎的，有时间、有耐心来干这种事情。

落不了地，落地很麻烦，也许有少数几个人可以做到，但是全面推广到一个大型的，20人，30人，50人，100人，200人的技术团队。很难的。

无招胜有招，没有什么所谓的code review的规范。

就是负责review的一定是高工及以上的人，高工至少是工作好几年了，少则工作3~5年，多则工作5~8年。技术专家，8年+。架构师，10年+。

你就以你的角度，和你当前的技术经验，技术的能力，对技术的掌握和把控，站在你的角度来审视别人的代码，如果你发现这哥儿们代码有些的烂的地方，你就给他提出code review的一些建议。

比如说你作为一个资深高工，你对23种设计模式怎么使用都有很深的体会。你看别人的代码的时候，突然发现有一段代码是一坨屎：

if() {

// 100行代码

} else if() {

// 200行代码

} else {

// 150行代码

}

方法长达500行代码

用策略模式+模板方法模式。。。。。。

你就可以找到那哥儿们跟他说，你该怎么怎么优化一下你的代码

代码一旦修改，同步的单元测试和冒烟测试，自己全部对应的做一下调整

静态代码扫描、单元测试覆盖率报告，全部重新做一遍

Pull Request再次提交到高工的手中，你再看一下当时让他修改的那些代码，发现，很好

853769620