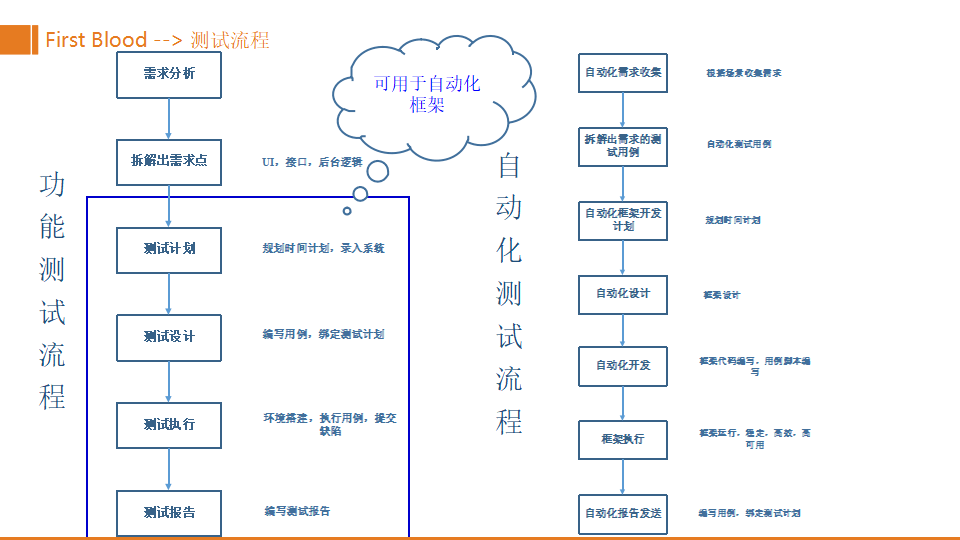
自动化测试流程



企业自动化测试人数：

功能测试人员：自动化测试人员 6:1左右 （针对于中小型企业功能和自动化组分开的情况）

中小型企业性能测试测试人员 1人

百度的情况（一个项目组5个测试人员，每个人都要会自动化，性能，功能）

自动化项目实际操作：

全新开发的项目，等到项目稳定之后（主干流程没有问题）开始介入ui自动化；ui界面问题较多之前，前期也可以做一些接口方面的自动化；

如果的迭代的项目，那么就会在svn里面吧原来的代码拉分支，到对应的分支版本；

原来老版本的代码和用例继续维护；新分支版本就去开发新的关键字或者函数；

自动化使用：

1.冒烟测试

开发提交新的版本，部署环境（自动部署），（ui、接口）自动化冒烟测试（一般是做主干流程），如果自动化冒烟测试无法通过，一般是查看自动化邮件结果，版本打回给开发人员；继续等待下一次开发提交测试。

自动化冒烟测试通过，一般是查看自动化邮件结果，那就就可以通知功能组做全面的系统测试；

2.功能回归：

老的一些功能自动化回归，新的功能人为手动测试；

3.线上验证：

做冒烟流程（主干功能流程）或者是checklist（功能测试人员确定下来的回归清单，每次都需要检查的一些功能点）回归：

自动化case筛选：

1. 首先收集主干流程，优先覆盖
2. 收集0级别的功能用例，按照优先级，能不能实现来优先覆盖
3. 最后才会去实现其他的低级别的case。直到慢慢覆盖
4. 一般的ui界面自动化case对应功能case覆盖率在40-60之间，界面变动较少的也可以覆盖到80%左右
5. 接口自动化可以覆盖到100%，除非很特殊的；

自动化在项目中的收益：

1.提高了测试组内效率（原来跑5条主流程需要1人2小时；目前自动化可以做到35分钟搞定）

2.提高精确率；（原来让人去做每个版本的功能回归，有人会偷懒，现在自动化就不会偷懒）

3.节省更多的时间（原来发版本都需要功能组人员加班到很晚，现在做自动化线上验证很快就搞定了）

4.覆盖率更高，提高质量（对于迭代版本，原来3天开发2天测试，时间来不及覆盖到所有的用例，那么就只会挑选重要的用例去执行；目前自动化就可以全部覆盖，边做手工的测试的时候，就可以边跑自动化）

5.提高组内测试人员技术（大家都会写代码，做自动化）

性能测试流程：

性能测试计划（预期时间 场景做性能）-设计脚本（事物，集合点，思考时间，参数化，关联）-

设计场景（真实、绝对并发，虚拟用户加载，场景运行多久）-基准测试（单用户运行的指标）-运行场景（监控资源，，）-结果分析