一、上线checklist

每个在本次要上线的版本中负责开发的同学，都需要提交自己的上线checklist，最终合并到同一份上线checklist文档中去。上线checklist文档，就是在里面要写清楚每个人负责的那个部分，那个子系统的整个软件工程流程过程中，沉淀下来的所有文档，都需要附加在这份checklist文档中，一般比如说可以用一个word文档，在这个word文档中，附加上你每个过程对应的一个截图和报告。

1、各系统的测试报告

1.1 XX子系统的测试报告

（1）单元测试报告：确认你的单元测试覆盖率是否达标

（2）冒烟测试报告：确认你的冒烟测试全部通过

（3）静态代码扫描报告：确认你的代码完全符合开发规范

（4）代码审查结果：就是在Gitlab中，你提交的PR最终会被人通过和merge，负责审核你的代码的人，他需要在审核之后，给出你一份总评，就是说，审核了你哪些哪些地方，然后最终判定通过，给一个截图。确认你的代码是经过审查的

1.2 XX子系统的测试报告

2、系统整体的测试报告

2.1 集成测试报告

2.2 系统测试报告

2.3 验收测试报告

这份文档，可以由架构师指派一个人去收集所有人沉淀下来的过程文档。一个大的需求版本上线，必须由这个系统的总架构师，亲自在上线前去审核这份文档，确保说，这个文档里反映出来的每个子系统都经过了良好和充足的测试，每个环节都做了，每个环节都按照标准、要求和规范去做了。

二、上线文档

架构师也可以指派一个人去做，有些公司，比如说一些较为传统的IT公司，一般对权限收的特别紧张，要求说只有经理才有级别去做什么代码合并，只有经理才有级别去操作线上系统，只有经理才有权限去执行上线。见过很多公司是这样子收缩权限的。

我的风格不是这样的。一般来说，我的理念，是说，尽量提升团队里每个人的能力。如果要提升每个人的能力，那么最好的办法，就是把他扔到线上的血与火的环境中去锤炼。有的人可能就工作个两三年那样子，甚至是一两年。但是如果他足够有潜力，你想要培养他的话，那就尽可能让他去做更多的事情。

尽量让年轻人做更多的事情

（1）详细设计：让组员自己去做，锻炼他的系统设计能力，而不是只会写代码

（2）工程初始化：让组员去做，锻炼他们从0开始迅速搭建出来一个系统的能力，而不是只会在现有的框架基础之上去填充代码

（3）集成测试、系统测试、验收测试：尽量让各个组员自己去做，配合QA、PM、其他团队的RD，让他们去协调，去沟通，锻炼他们的跨团队的沟通协作能力

（4）系统上线：让年轻的成员去进行线上操作，这样可以真正锻炼他们的能力，让他们有线上操作和运维的能力

如果高工走了，可能就没人能接活儿，没人去做设计，没有人会从0开始搭建一个工程出来，没人能协作组织跨团队的测试协作，没人能执行系统上线，和线上运维的一些操作

可能就导致说，你作为一个架构师，很累，什么事情，都要事必躬亲

1、初始化线上数据库

（1）从哪儿导出来一份SQL文件

（2）在哪个线上数据库中执行这份SQL文件

（3）执行过后，需要检查一下，需要的128张表是否全部完成初始化

2、部署系统到线上机器

（1）将完整的代码，打成war包

（2）到线上的哪台机器上去，将tomcat停止

（3）然后用scp将war包上传到机器上去，然后放入tomcat的webapp目录下

（4）然后重新启动tomcat服务器

（5）观察系统启动日志，各个环节的初始化是否正常，比如说数据库连接池的初始化

3、线上验证

（1）系统成功部署之后

（2）对几个核心流程和功能，手动执行一些操作

（3）确认所有功能正常运行

（4）确认所有的日志都正常打印

（5）确认所有的数据库中的数据记录都正常

三、执行上线

执行上线，一般会有一个规范，就是在什么时间点，可以执行上线

一般来说，是选择系统的低峰期

分开来说，不同的系统上线的情况，可以允许的时间是不一样的

（1）大版本，上线，v1.0，v1.2，一般是建议在晚上9点以后，就是在低峰期，甚至是建议在凌晨2点，凌晨5点

（2）小版本，修复一个bug，做了一些改动，一般是建议在非高峰期的一些相对低峰的时间段，可以执行上线，上午的10点以前，下午2点~4点

执行上线的规范

（1）需要超过2个以上的人在场

（2）一般是负责执行上线的同学，按照之前审核过的上线文档来一步一步执行操作

（3）级别较高的同学，高工，会去在旁边仔细看着他上线的每个步骤

（4）如果一旦出现任何问题，需要进行回滚，比如说，用之前版本的代码，再次重新上线