

测试左移

1

是什么？

测试左移的思想本质是越早的发现不合理的地方，出问题的几率就越低。测试左移的原则支持测试团队在软件开发周期早期和所有干系人合作。因此他们能清晰地理解需求以及设计测试用例去帮助软件“快速失败”，促使团队更早的修改所有的bug。

参与和理解会使测试人员获取产品完整的知识，彻底想清楚各种场景，根据软件行为设计实时的场景，这些都会帮助团队在编码完成之前识别出一些缺陷。

2

怎么样？

从不重视代码质量的第一天开始，就埋下了问题修复，定位的成本和修复问题再次引入问题的成本。当测试在周期的早期开始时，团队会更专注于质量，并且“让我们在第一时间获得正确的编码”前景。这有助于节省大量时间，并减少软件开发团队必须为特定代码执行的迭代次数。因此需要提高质量上限和质量下限。

提高质量上限：通过一系列活动，来避免问题或者本身就让我们起步就变得很好的，一句话：良好的开始是成功的一半。

提高质量下限：通过一系列的活动，让我们的质量成果得以保证。

测试左移，其实就是通过一系列的活动，能提高质量的上限，缩短测试的周期，提高质量的下限，这样我们就可以在不断提高下限的过程中，始终将质量稳定在一个水平线上，而和团队一起追求更高的目标。

3

怎么做？

在团队的Devops开发下，对于测试左移进行的操作：

- 编写单元测试，通过单元测试提前进行测试；
- Code Review，通过代码走读发现一些基础的问题；
- 参与需求评审，提出需求不清晰、不合理、遗漏等意见，了解开发的实现方式；
- 参与研发需求分解，协助梳理分解遗漏点；
- 参与概要、接口设计评审，协助梳理遗漏逻辑；
- 提早输出测试导图，开发编码前进行评审；
- 部分功能提测，提早开始测试；
- 自动化测试，用于回归确保旧版本功能正确性；

对于测试左移，进行了相应的尝试后，也发现了测试左移实践的问题：

- 测试要求提供概要设计、接口文档；
- 测试要求单元测试必须通过；
- 测试干预需求设计；

很多人都认为是测试在要求完成一些没必要的事情，测试在干预我的工作。

其实问题的矛盾点在于前面说过的一句话：不管是测试左移还是测试右移，都是为产品质量服务。不要把提测认为是测试活动的开始，上线是测试活动的结束，更不要认为质量只是测试人员需要关注的。对于测试左移的落实，最重要的就是全员质量服务意识的培养。

4

测试需要关注什么？

对于测试左移其实我们还有很多东西要做，就好像前面说到的都是为产品质量服务，那么在研发流程中的任何角色、人员都要为质量服务。

- 了解和评估需求，发掘需求的不合理之处，提供针对性的建议，提高需求的质量。
- 了解和评估产品设计方案，发现设计的漏洞，并及时反馈给产品人员，避免问题到了测试阶段才暴露出来，影响产品按时发布。
- 了解产品的技术实现，通过查看开发文档，接口文档等，执行静态测试发掘问题，检查代码问题。

提高质量上限：

- 健康的项目流程（合理并且严格遵守的项目流程）；
- 合理的需求分析（评估需求的质量，分析需求的合理性以及完整性）；
- 出色的系统架构；
- 充分利用静态代码扫描；
- 进行研发标准的定义；

提高质量下限：

- 健康的测试流程；
- 优秀的测试用例；
- 合理的测试计划；
- 合适的自动化；
- 适当的探索式测试；
- 开发自测（TDD、BDD，测试提供更好的用例、技术支持）；
- 尽早的测试；
- 团队质量意识的培养；

对于测试左移，也需要一个重要的基础，工程习惯，SDLC成熟度，测试分层，持续集成，链路上延展发布的节奏，纵深上需要贴合业务的专精领域的深度探索，代码扫描（规范，问题，安全，异常等），CR，代码提交行为分析，test double（mock，fake，stub，dummy），UT，自动化，验收测试等。左移需要工程效率具备不亚于研发的代码能力。因此对于测试左移，可以围绕质量服务思想展开，参与人员则不仅仅局限于测试人员。

测试右移

1

是什么？

左移是往测试之前的开发阶段移，右移是往发布之后移。也就是产品上线了之后也可以进行一些测试活动。当然在生产环境直接做测试是不推荐的，但是我们可以**在生产环境做监控，监控线上性能和可用率，一旦线上发生任何问题，尽快反应，提前反应，给用户良好的体验。技术人员要比业务方先发现问题，如果业务方已经发现业务量明显下降，说明问题已经很严重了。**

测试右移其实还可以理解为如果线上发生任何问题，我们有没有能力第一时间发现问题并解决问题，并保证线上数据的一致性或尽可能少的影响线上用户，以及并且实时获取用户反馈。

2

怎么样？

实践起来也是存在问题，除了技术问题之外，还有例如：

- 线上监控搭建后使用率不高；
- 线上问题反馈机制，业务人员不配合等等；
- 监控指标不合理，反而被认为增加服务器负载；
- 测试右移的落实，除了质量服务的培养，更加重要的反而可能是：完善的反馈、发现、定位，在监控-架构完善后，怎么更好的与项目工作(流程)结合，不要让其成为累赘。

3

怎么做？

对于测试右移，线上监控可以是突破点：

- 闭环的线上问题反馈-检查-解决-更新流程；
- **更便捷的日志查看、回传服务；**
- **丰富有效的log，便于问题的快速定位；**
- **丰富的监控指标（例如业务异常点指标）；**
- 成本监控（例如短信发送等）；
- 关键指标每日监控（服务器指标）；
- **生产数据监控（警报）**（通过sql语句实现生产数据监控，例如是否有多个订单号一样的订单出现等）；
- 因此对于测试右移，可以围绕问题反馈、发现、定位、监控展开，参与人员则不仅仅局限于运维人员。

4

测试需要关注什么？

软件一经发布，就要对线上日志监控和预警。及早发现问题，及早修复，将损失降到最低。而不是等用户反馈了，才去解决。如果等到用户反馈问题，说明问题影响反馈已经很大了。

- **测试上线及时验证，有问题，开发快速回滚代码**
- **上线后开发监控服务日志，日志报错，代码回滚**
- **xdcs监控服务流量，出现流量报警快速定位问题**
- **关键指标每日监控（服务器指标）**
- **生产数据监控（警报）**
- **用户反馈问题及时跟进，针对缺陷，通知开发尽快解决，针对体验，通知产品打磨细节，更好的服务用户。**

灰度发布，数据隔离，线上监控，线上验收。当业务方由于环境等原因，**有些功能必须在线上验收。**线上验收的原则是尽可能的不要影响到原有功能和使用业务的用户，这个就需要做好很好的隔离，所

以从研发一开始的设计就从线上可测性角度就需要考虑到这一点，功能做好隔离，数据做好隔离，一旦出现问题，我们有相对应的风险预案，如何清除脏数据，如何将功能降级等，前期的设计都要考虑好，发布完成以后我们还需要考虑运营层面的事情。

采集线上数据，包括业务数据和性能数据，建立一个健康的产品数据模型。业务数据可以通过数据挖掘，生成用户画像，有助于产品人员做出相应的决策，包括改进现有功能，开发新功能满足用户需求，砍掉某些不合适的功能等。性能数据，最直接的作用就是运维人员做出相应的硬件调整，包括硬件扩容，整合闲置资源等。也可以给架构师提供一些系统架构设计上的调整。

四

总结

测试左移的落实，最重要的就是全员质量服务意识的培养。测试右移的落实，除了质量服务的培养，更加重要的是：完善的反馈、发现、定位问题，提升用户体验。相比较，测试左移的价值更高，尽早发现并解决问题，成本更低。

无论是测试左移还是测试右移，测试人员开始从质量控制转变成质量保证，从被动发现软件问题到主动提高软件质量，从团队的边缘人物转变成团队的活跃分子，这都值得为之践行。