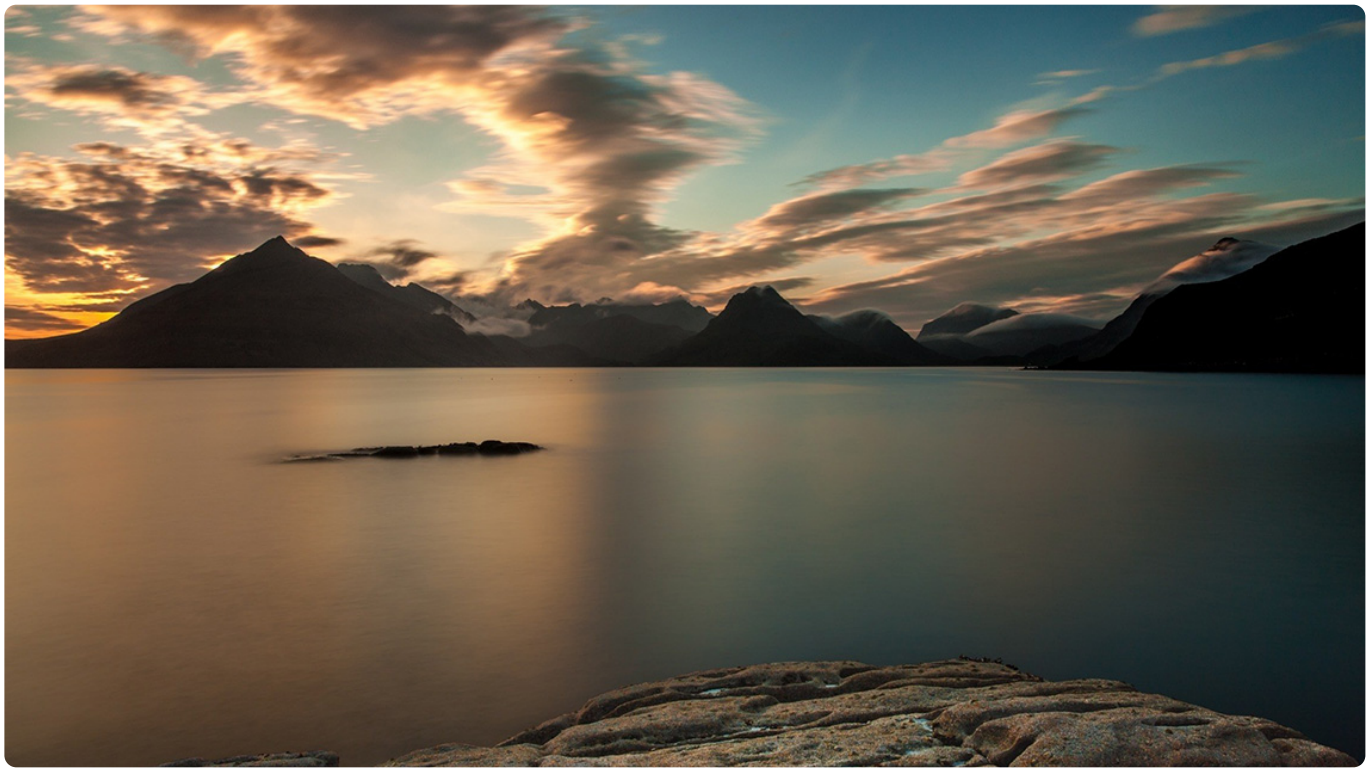
第55讲 | 用机器打造迭代机器：现代研发流程体系打造（二）

2018-07-18 爱范儿CTO兼知晓云负责人何世友

技术领导力300讲 进入课程 >



讲述：黄洲君
时长 10:17 大小 4.72M

你好，我是爱范儿 CTO 兼知晓云负责人何世友，今天想跟大家继续聊聊“打造现代研发流程体系”这个话题，并将着重跟大家分享其中“用机器打造迭代机器”这一部分内容。

在上一篇文章里，我们分析了研发流程中的关键环节，并给出了对应的解法。它们分别是——

1. 高速运转的传送带

现代化的项目管理（任务流转）工具。

2. 可追溯的迭代

通过传送带，将每一次迭代的产物，如代码提交、架构设计变更、测试构建部署等串联并存储起来。

3. 重要角色的沟通

用一个通用平台，如 Slack，在解决人与人之间通讯的基础上，重点解决系统工具与人之间的沟通问题。

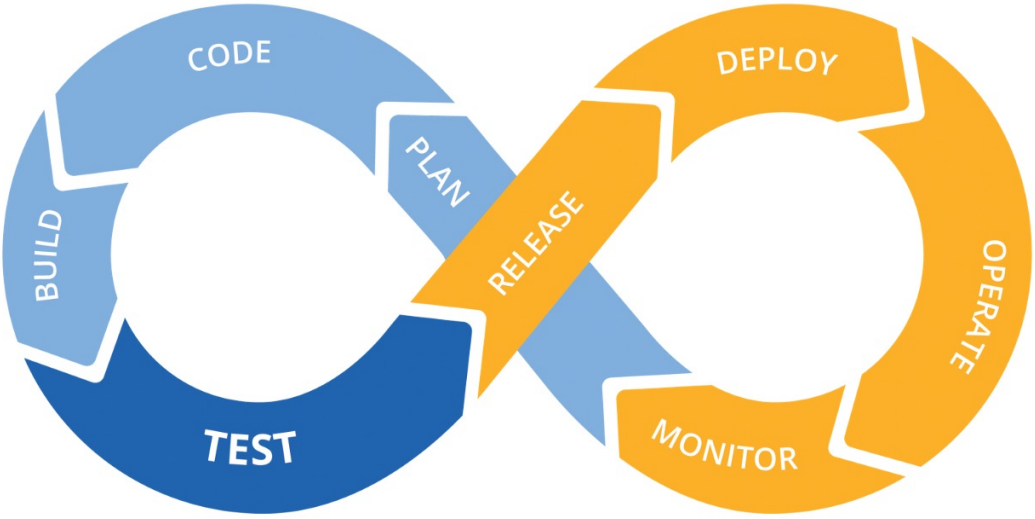
4. 用机器打造迭代机器

受限于文章的篇幅，上篇文章中只是简单说到了因为迭代的步骤很多，所以要让机器包揽大部分环节，估计很多读者并不能十分感同身受。本文将对此做详细解释：为什么要用机器打造迭代机器？

迭代频率越高，对迭代里的自动化程度的要求就越高。打个简单的比方，如果项目要求一天迭代两次，测试工程师就要一天走完两次主流程回归测试。此时，人工就是最大的瓶颈。一个项目分分钟有成千上万个用例，依靠有限的测试人员分拣完成，那就是纯体力活了。而对质量的要求越高，主流程的覆盖范围就越广。单就这一个环节，如果没有机器的参与做自动化，就会成为一个不可调和的瓶颈了。

之前提到，构成自动化流程的大部分工具都是现成的、可以花合理价钱买到的，本文就将重点介绍研发流程里的各种工具们，以及不同场景下的具体选型。由于这些工具被正确配置完成之后，拥有脱离人工干预在不断电的情况下自我运转的能力，我们亲切地称之为迭代机器里的机器们。

迭代机器里的流水线



编码【人类才智】⇒代码审查【人类才智】⇒静态检查【机器】⇒单元测试【机器】⇒测试【机器】⇒构建【机器】⇒部署【机器】⇒监控【机器】⇒自动扩（缩）容【机器】。

这是一个环环相扣的流水线，每一个步骤都由一个机器角色完成并推送到下一个步骤，最终完成全程；一旦其中一环无法完成，则本次迭代就宣告失败，需要返工。

通常这样的流水线跑在 CI 系统上。CI 系统，Continuous Integration，持续集成系统。常见的 CI 有 Jenkins、Bamboo、Solano CI 等。这些系统各有千秋美丑，本文不赘述对比，各位可根据团队背景和成员喜好进行选择，只需要看这个系统是否具有真正意义上的可定制拓展性，以及规模可观的第三方服务的接入支持。

下边将围绕流水线中的几个重要环节进行描述。

静态检查、单元测试、构建

静态检查、单元测试和构建环节发生在每一次代码提交之后，由代码版本库的代码提交事件触发执行，如 Github、Bitbucket、Gitlab 的 Webhook 等。

1. 静态检查

静态检查由各语言的语法 Lint 工具和 bug 检查工具组成，前者包括 JS 的 eslint、Python 的 pylint 等，后者包括 Java 的 findbugs 等。

2. 单元测试

基本上每种语言、每种框架都有支持单元测试，如 JS 的 Mocha、Python 的 unittest、Java 的 JUnit、Go 的 testing 等。工具本身都是比较类似的，各语言开发者都应该很熟悉。难在代码编写时的测试用例覆盖，这是一个需要仔细权衡覆盖度和时间的过程。

在实践中，比较推荐的是，让测试工程师参与到单元测试编写中来，每一次项目的测试用例评审，一部分用例一定要转化为开发工程师的单元测试。或者说，测试工程师需要参与到开发工程师的单元测试审查和覆盖率评估中，对应的，开发工程师也要参与到测试用例评审中。这二者相结合，黑白盒、单元集成测试才能真正有机的为项目质量负责。

3. 构建

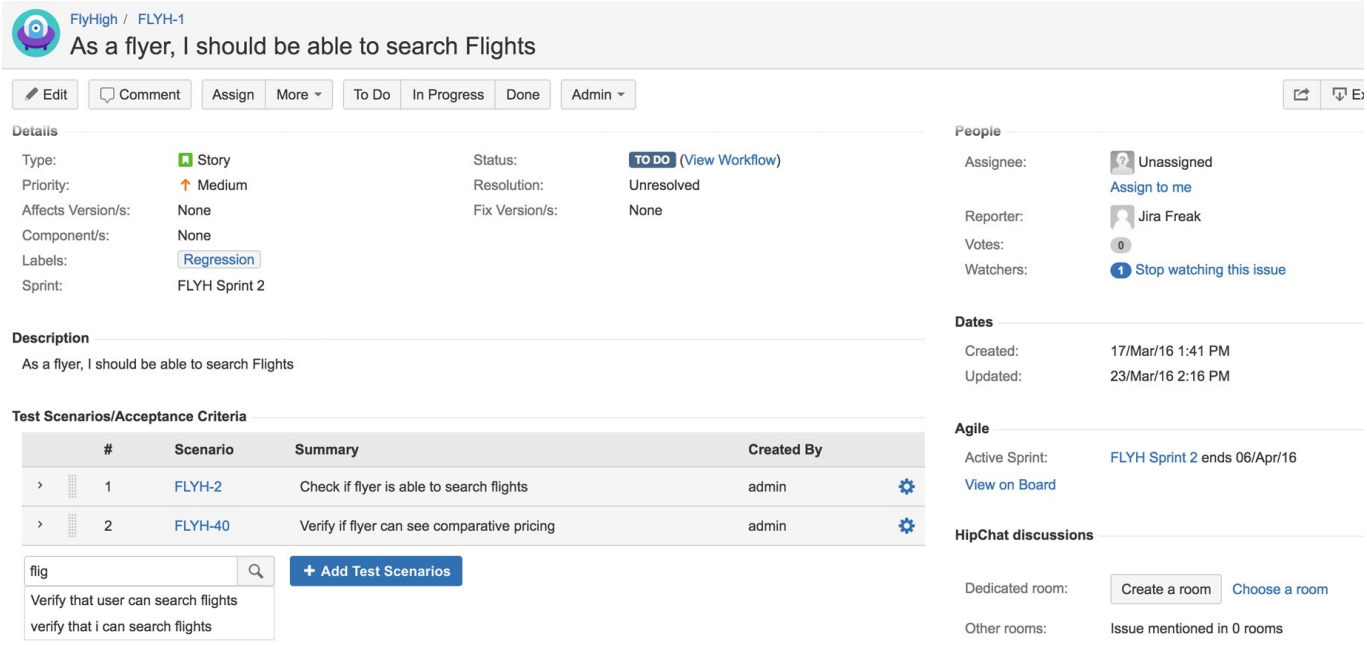
构建（Build）是需要开发工程师根据项目的部署策略编写对应的构建打包脚本。例如前端的 webpack、后端的 maven、客户端的打包等。不过基本上要做的事情就是将原本由工程师在本机上跑的脚本移植到 CI 系统上，这个过程本身不带来多少成本。

自动化测试

自动化测试由测试工程师维护，由两块工作构成：测试用例管理、自动化测试脚本编写。

1. 测试用例管理

测试用例管理流行的工具有 Excel、Qmetry、TestLink 等。没错，Excel 在众多公司中依然是管理测试用例的好工具。当然在我们讨论的场景里，Excel 已经不堪其用。管理测试用例，是为了让测试用例和对应的需求描述，也就是 User Story 关联上。从而让每一轮测试执行，也就是 Test Run 的结果，不论是成功或失败，都自动回传到关联的任务状态里。



同时，平台级的用例管理，可以让用例迭代起来，和项目一起生长，这和前文提到的架构设计文档的迭代一脉相承。甚至，用例要作为项目推进中的自动化测试的组织枢纽，串联起自动化测试和实际任务的状态流转。

2. 自动化测试

这里的自动化测试，主要指的是黑盒测试和集成测试，和开发工程师维护的单元测试做一个区分。常用的框架有 LoadRunner、Selenium、Appium 等。这是一个十分耗时耗力的过程，常见的做法是梳理经过每一次迭代的测试用例，最终形成一个主流程用例集。

主流程的特征是产品特性基本稳定，不会在短期内有较大改动。测试工程师需要对主流程的测试用例进行测试覆盖，例如通过 Selenium 进行 UIUE 的用户交互过程录制等。而有了主流程的覆盖，每次迭代的发布，才能够真正的做到 push on green。否则，每次发版还得测试人员手工回归确认，那基本就是一个跨不过去的时间鸿沟了。

部署、监控、自动扩（缩）容

1. 部署

Devops 的兴起让运维得到解放，Docker 的流行也让社区疯狂，似乎非 Docker 不可，不上容器不是好技术团队。其实不见得。重要的是达到目的，而非工具，一定要根据项目实际情况进行技术选型，不要因为一种便利引入额外的麻烦。

目的是什么？目的是让机器自己完成自动化部署。

而通过前面介绍的工作，CI 流水线上已经有了经过完整测试的构建产物，于是部署阶段只剩下：

1. 开启并初始化机器，并完成系统环境配置，通常可以用预先准备好的镜像文件完成该步骤；
2. 上传构建产物到机器上，启动服务；
3. 将流量或任务分发到新的机器上；
4. 下线旧的机器。

这里边有机器的运维、服务的部署、负载均衡器配置等，每一项业内都有非常不错的工具可以用。比如我们在爱范儿目前用的是——

1. 使用 Ansible 完成环境的自动配置，结合 AWS 的 ami 镜像完成机器运维部分；
2. 使用 Fabric 结合 AWS 的 CodeDeploy 完成的部署流程，并在此基础上完成了 Auto Scale。

AWS 的 CodeDeploy 非常好地利用了 AWS 的基础设施，可以一键完成上面提到的 1、2、3、4 这几个环节。而在使用 CodeDeploy 之前，团队写了一系列脚本去做这块的工作。

Auto Scale 常见的做法是在监控系统里定义一系列的 Metrics，设定阈值，比如，“过去 2 分钟内机器 CPU 达到 60% 以上”就是一个完整的阈值条件定义。然后为这个阈值配置一个动作，比如“过去 2 分钟内机器 CPU 达到 60% 以上，部署并上线 4 台新的应用服务器”。而怎么部署并上线新机器就是上文的 CodeDeploy 的活儿了。

2. 监控

监控分异常监控和性能监控。

异常监控配合日志收集器工作，如前端的 Sentry，后端的 Cloud Watch（AWS）、loggly 等。基本的工作原理是从日志中获取错误信息，并进行统计，达到设定的阈值就开始报警。异常监控系统经常对接的是电话告警系统，为 OnCall 的工程师提供错误叫醒服务。

性能监控分微观和宏观两个维度：

1. APM 探针实时收集应用服务器代码层面的性能信息；
2. 机器状态（cpu、mem、load）、应用服务响应时间等。

这两者结合可以无死角反映服务状态，对接到 Auto Scale 系统后可以在大多数情况下完成自动化运维。

APM 探针服务，常用的有 NewRelic、AppDynamics 等，国内也有听云、OneAPM 等一众厂商。区别基本上就是价格和第三方支持完备度，近些年各大云服务商也在做这些周边支持，选择上并不是难事。也有一些团队将机器状态等信息归到异常监控里，而我们归到性能监控，主要是站在能否让机器自己治理的角度。毕竟，放到异常监控，OnCall 的工程师更容易醒，但醒了不也是要做 Scale 的事情嘛。

说到这里，大家在迭代流水线中的各个环节上都用了哪些工具呢？欢迎在留言中告诉我们，供大家参考。

一些待完成的和待思考的

然而，一台高速运转的迭代机器中，人类角色才是真正的瓶颈。或者说，高速迭代对人员的要求会更高。因此，团队的成长便更为重要，也更值得探讨。坊间有一句话，互联网公司都是轻资产，值钱的就是人、流程、代码。

那如何打造一个高速成长的团队呢？希望有机会可以和大家探讨。

作者简介

何世友，爱范儿 CTO TGO 鲲鹏会广州分会董事会成员，学习委员。从校园创业到跨国团队技术顾问再到如今，专注于高并发网络、机器学习、移动 APP（部分硬件）、团队管理。

上一篇　第54讲 | 打造高速运转的迭代机器：现代研发流程体系打造（一）

下一篇　第56讲 | 有了敏捷开发，那交付期限去哪儿了？

精选留言 (4)

写留言

wenhao
2019-05-15

没有介绍传送带了。用什么工具可以把各环节串联起来，需要自己开发吗

展开

Ernest 何...
2018-07-18

@walkingdonkey，感谢指出，这里刊误下：黑盒。

展开

walkingdon...
2018-07-18

"这里的自动化测试，主要指的是白盒测试和集成测试"，是不是应该是黑盒测试，而不是白盒测试?

walkingdon...
2018-07-18

"自动化测试，主要是指白盒测试和集成测试"，是不是说错了？应该是黑盒测试，不是白盒测试吧?