

澳洋医院 CI/CD 平台建设方案

张旭皓

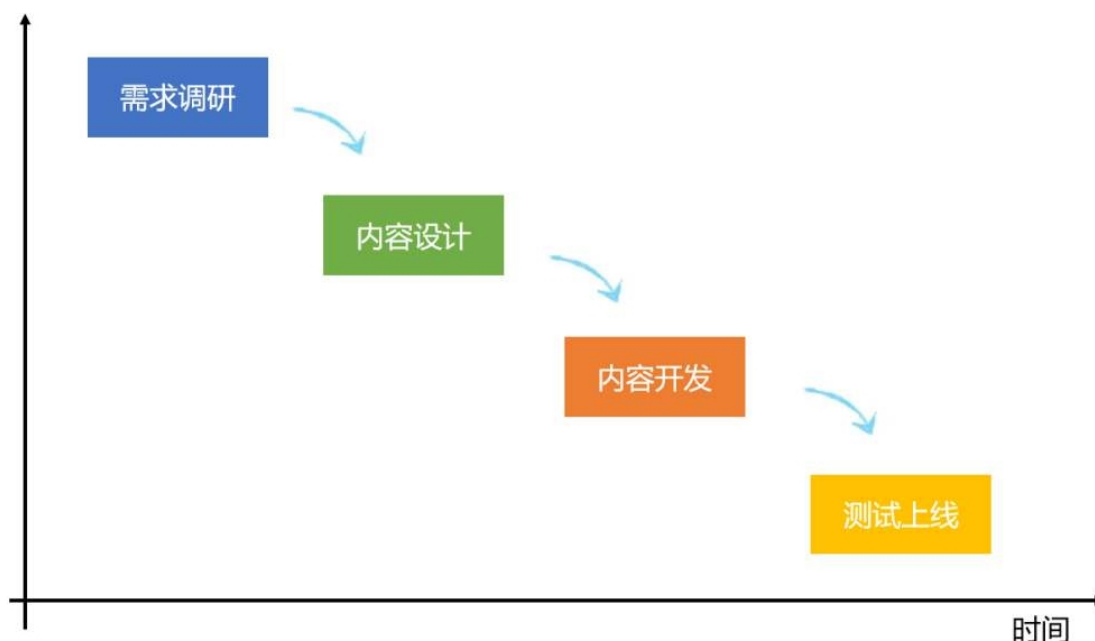
2019/8/15

澳洋医院健管组

一、引言

在讲到 CI/CD(持续集成/持续部署)之前,先要讲下 3 种开发模式瀑布式开发/敏捷开发/DevOps。

1.1 瀑布式开发



瀑布模型式是最典型的预见性的方法，严格遵循预先计划的需求分析、设计、编码、集成、测试、维护的步骤顺序进行。可以从上图看出，时间的成本会随着过程的推进而增加，其中需求有一丁点的变动就要从中间往后按步推进，适合人员较多的团队，开发，测试，运维人员都具备的情况，工作量可以细分，项目可以有条不紊的推进。

瀑布式开发的好处在于人员的合理安排，工作的合理安排，项目按进度推进，有条不紊的走下去。但是随着现代开发过程中，瀑布式开发出现了它很多的缺点，首先是大量的文档维护:在瀑布式开发中，因为开发，运维，测试，需求等多方面的人员参与其中，文档的维护肯定是要有的，项目调研文档，项目需求文档，开发文档，api 接口文档，测试文档，运维文档，交付文档等等，非常多的细节文档需要去放入项目库中维护项目的完整性，以避免中间一块人员的变动导致项目的进度无法

开发过程是线性的，成果的验收只能等待整个过程结束后才能预见到。其实现实过程

中根本不可能，就以健管系统为例，开发的过程中走的也是瀑布式开发模型，上线后总会有大大小小的问题存在，还有些需求的变更，以及一些业务人员的操作方式等测试过程中无法测试出来的情况，没人哪个人可以拍着胸脯说自己写的程序没有 bug，敢这么说的他基本没干过程序员。

需求的变更也是瀑布式模型无法适应的一点，小的变更其实还好，大的变更按照瀑布式模型的走下去，时间人员成本也会随之增加，所以老生常谈的外包公司，钱到位，什么需求都可以改，这个梗 csdn 之前都有聊过。

1.2 敏捷开发

敏捷开发（Agile）是一种以人为核心、迭代、循序渐进的开发方法。敏捷开发以用户需求为核心，采用迭代(时间周期)、增量(循序渐进，功能模块)的方式开发软件，目的在于快速覆盖、响应市场需求。

大项目划分为小项目，分别完成，独立运行，如微服务的开发过程，就是将系统独立进行开发。

敏捷开发的特征如下：



迭代式开发 项目按照时间周期进行迭代，比如 A 功能优先级比较高，则在第一个迭代周期内优先开发 A 功能，并上线。第二个迭代周期开发 B 功能。

增量交付 敏捷开发注重的是迭代式开发，相比瀑布式的漫长的开发，一个模块一个模块的交付到客户使用。现在的需求变更的是非常的快速，瀑布式开发不一定能保证客户的满意度，有可能开发到一半客户就会变更需求等。比如微软在吸取 Vista 操作系统采用传统的瀑布式开发流程之后，发现操作系统并不能完全覆盖用户的需求。在 2005-2007 两年时间内 通过内部推行的敏捷开发原则，上线了 win7，获得市场的一致好评。

采用敏捷开发的产品在产品初期会上线基本功能，之后的功能是根据收集到的用户反馈进行开发的，实现功能模块的持续集成。

持续集成 采用敏捷开发的产品在产品初期会上线基本功能，之后的功能是根据收集到的用户反馈进行开发的，实现功能模块的持续集成。比如说健康管理体统小组 3 天一小改发布,5 天一大改发布。小改发布不影响线上业务的运行，大改操作需要关闭服务器停止线上业务运行进行发布。

开发团队和用户反馈推动产品开发 敏捷开发提倡用户参与到产品或项目开发的整个流程当中，通过用户反馈使得产品更加符合用户频繁变动的需求。比如说我直接接过的一个修改需求，需求是提高医生添加诊断建议的速度，提高体检医生的业务量，在初期的完成后我往往是推到测试环境进行自我的测试，开发人员的测试和单元测试完成后，我会和主/预检医生，以及健管系统的主管约定时间进行业务人员的测试，通过她们的实际使用情况来确保我做的这个达到该需求的验收标准。经过她们提出的一些小的建议以及改动后确认无误后发布至正式环境供所有的医生使用。软件毕竟是业务人员的使用，他们方便了才是真的方便了。

开发团队自我管理 传统的开发模式，注重文档约束，而敏捷开发原则的推行原则要求团队内部交流便利、文化相对开发，除去必要的文档约束，如 Api 接口文档，最注重的是团队成员的高效交流，以此来提高产品、项目的开发效率、开发质量。通常在一个模块结束后我会抽时间整理对应的设计文档，api 接口文档放入项目的 svn 托管地址中供所有团队成员参阅，周六半天的学习日是团队交流的好时机，或者是修补项目重大 bug，项目中技术的困难点，都可以在周六学习日进行交流，一方面充实了自己也充实了团队增长。

1.3 DevOps

DevOps（Development 和 Operations 的组合词）是一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。从定义来看，其实 devops 就是为了让开发、运维和 QA 可以高效协作的流程。说白了也就是开发运维兼一体。



其中 DevOps 推崇的一个自动化

强大的部署自动化手段确保部署任务的可重复性、减少部署出错的可能性。

与传统开发方法那种大规模的、不频繁的发布（通常以“季度”或“年”为单位）相比，敏捷方法大大提升了发布频率（通常以“天”或“周”为单位）。

二、CI/CD 概述

2.1 CI 持续集成

现代应用开发的目标是让多位开发人员同时处理同一应用的不同功能。但是，如果企业安排在一天内将所有分支源代码合并在一起（称为“合并日”），最终可能造成工作繁琐、耗时，而且需要手动完成。这是因为当一位独立工作的开发人员对应用进行更改时，有可能会与其他开发人员同时进行的更改发生冲突。如果每个开发人员都自定义自己的本地集

成开发环境（IDE），而不是让团队就一个基于云的 IDE 达成一致，那么就会让问题更加雪上加霜。

持续集成（CI）可以帮助开发人员更加频繁地（有时甚至每天）将代码更改合并到共享分支或“主干”中。一旦开发人员对应用所做的更改被合并，系统就会通过自动构建应用并运行不同级别的自动化测试（通常是单元测试和集成测试）来验证这些更改，确保这些更改没有对应用造成破坏。这意味着测试内容涵盖了从类和函数到构成整个应用的不同模块。如果自动化测试发现新代码和现有代码之间存在冲突，CI 可以更加轻松地快速修复这些错误。

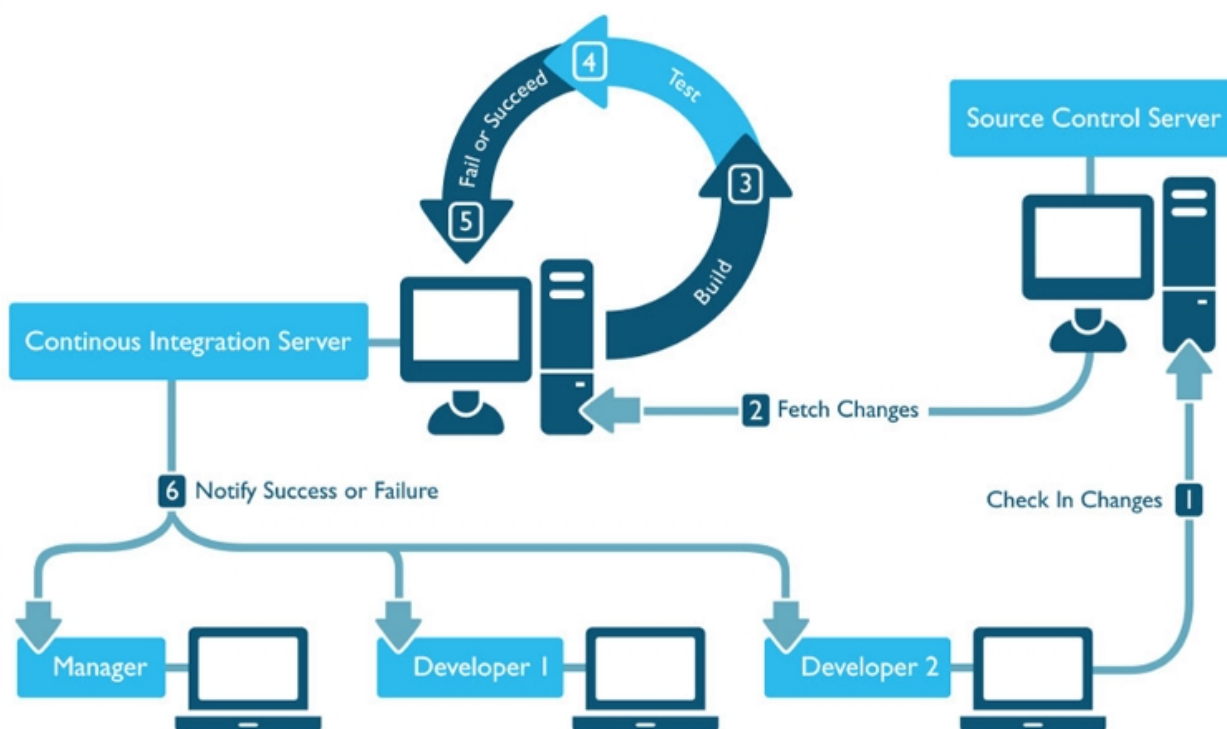


图 典型的 CI 流程

2.2 CD 持续交付

完成 CI 中构建及单元测试和集成测试的自动化流程后，持续交付可自动将已验证的代码发布到存储库。为了实现高效的持续交付流程，务必要确保 CI 已内置于开发管道。持续交付的目标是拥有一个可随时部署到生产环境的代码库。

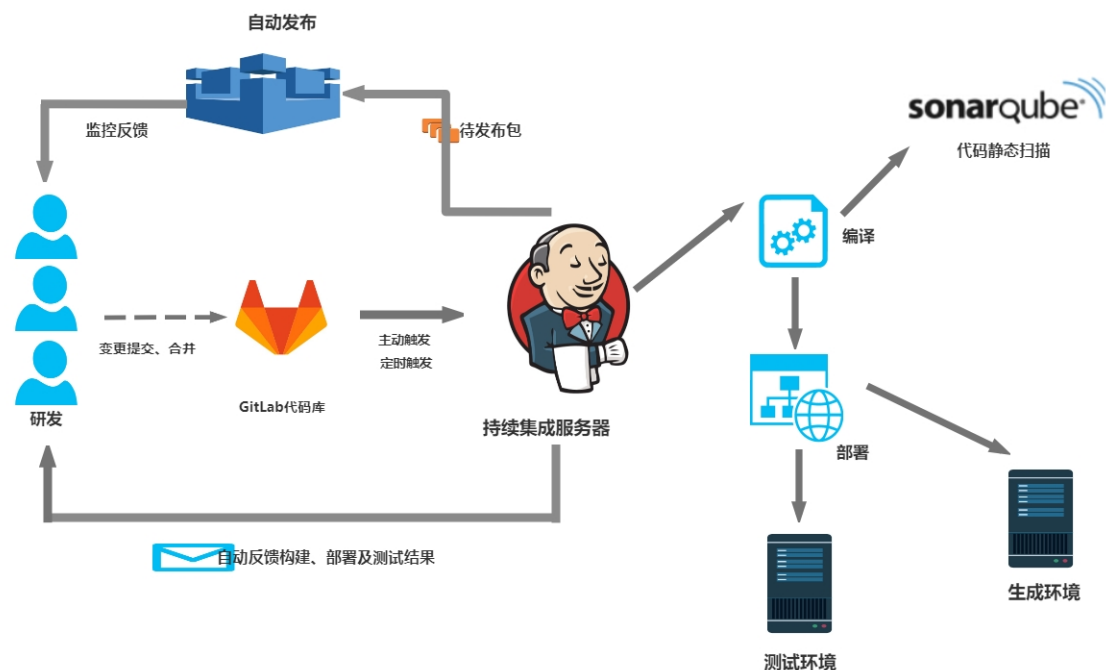
2.3 CD 持续部署

通过自动化部署的手段将软件功能频繁的进行交付。与持续交付以及持续集成相比，

持续部署强调了通过 **automated deployment** 的手段，对新的软件功能进行集成。相比传统的人工部署而言，自动化部署可以减少人工带来的出错率。

三、澳洋医院 CI/CD 架构图

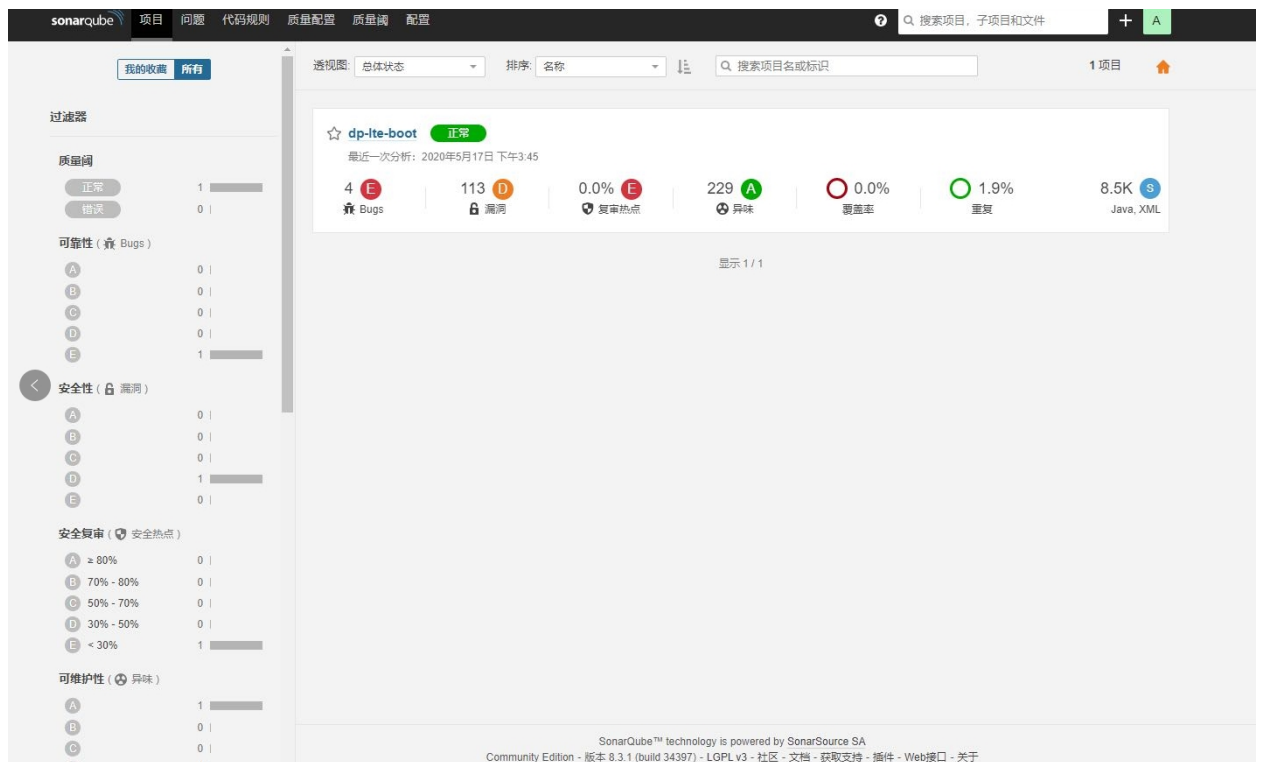
结合线上的一些实际情况，手动式的发包以及测试的不完整性导致线上的问题频繁出现，为了解决以及监控这些问题，于是想到了 **CICD** 建设，对于企业而言，**CICD** 的建设上之后对企业或是开发人员都是好处颇多，不仅可以把复杂的发布流程简化，还可以进行代码质量的扫描帮助软件的重大漏洞。



3.1 SonarQube

SonarQube 是一款用于代码质量管理的开源工具，它主要用于管理源代码的质量。通过插件形式，可以支持众多计算机语言，比如 java, C#, go, C/C++, PL/SQL, Cobol, JavaScrp, Groovy 等。sonar 可以通过 PMD, CheckStyle, Findbugs 等等代码规则检测工具来检测你的代码，帮助你发现代码的漏洞，Bug，异味等信息。

SonarQube 可以自定义代码规范内容，结合使用阿里 P3C（俗称的 java 开发规范）可以管控项目的重大 bug 等



为什么使用 SonarQube?

到目前为止，没有哪个 CI 工具有一个插件可以将使所有上面这些工具集成到一起。没有哪个 CI 工具可以提供良好的钻取功能，但 sonarqube 有。没有 CI 插件可以将所有的软件质量的度量数据整合到一起。没有 CI 插件提供管理视角。还没有设计/架构问题相关的 CI 插件 没有 CI 工具或插件提供整体项目质量的仪表盘。

3.2 jenkins

一句话来概括 Jenkins 工具：它是一个软件在生产过程中持续构建，持续发布，持续部署等 devOps 行为的代理工具。

回顾下传统软件发布的一些流程:

1. 拉去最新的代码
2. 编译/打包
3. 上传到服务器
4. 服务器进行配置
5. 部署软件测试软件

一旦代码频繁的更新，那么以上流程将会被反复执行一次甚至多次。有了 jenkins 之后，部署将会变得更简单,出错率也会变得更少。对于多机器的部署，有了 jenkins 之后部署也

会变得更轻松。

四、总结

自动化云运维时代已经来临，上面的只是非常基础版的 CICD 架构，满足现有的健康管理系统的频繁的需求变更而萌芽出 CI/CD 平台的建设，一方面不仅将繁琐的发布流程给简化了，另一方面开发人员可以自己手动一键部署到测试环境进行测试，不需要专门的人员进行打包上传到服务器测试，还可以监控到项目的质量阀，可以对项目的代码质量有着一个更好的管控。

DevOps 时代已经来临，合理的利用工具和平台可以解放我们的双手去处理更多跟业务有关的事。

