

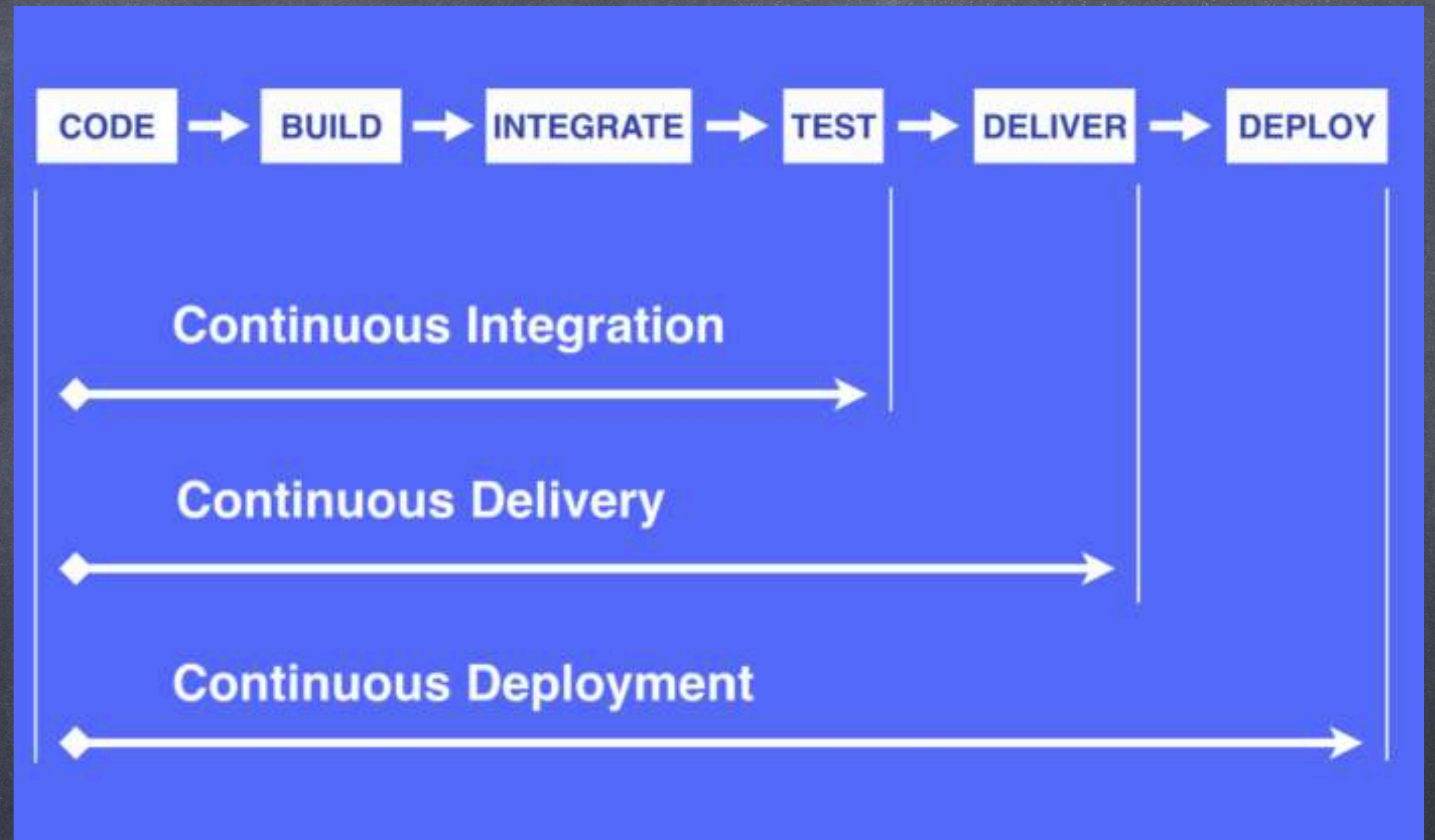
前端工程化之CI/CD

京程一灯

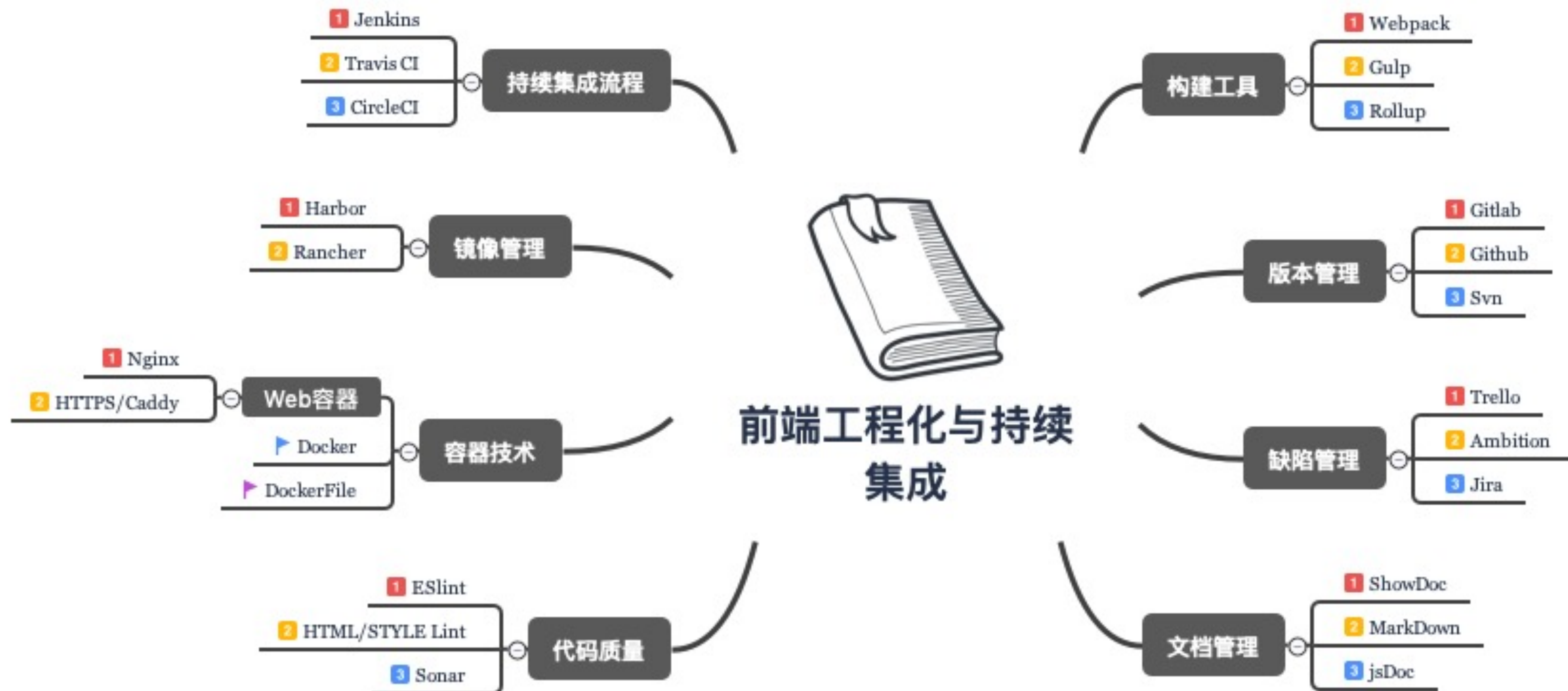
<http://www.yidengxuetang.com>

课程概要

- CI/CD概述
- 自动构建流程
- 版本控制
- 持续集成 (CI)
- 持续交付和持续部署 (CD)
- Jenkins与Travis



全景图

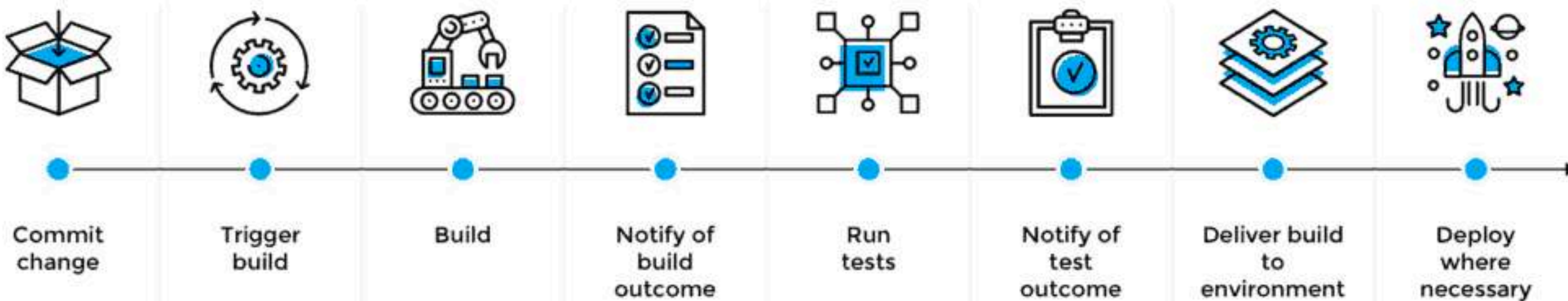


CI & CD

- 持续集成 (CI)
- 持续交付和持续部署 (CD)
- 现代软件开发的需求加上部署到不同基础设施的复杂性使得创建应用程序成为一个繁琐的过程。当应用程序出现规模性增长，开发团队人员变得更分散时，快速且不断地生产和发布软件的流程将会变得更加困难。
- 为了解决这些问题，开发团队开始探索新的策略来使他们的构建、测试和发布流程自动化，以帮助其更快地部署新的生产。这就是持续交付和持续集成发展的由来。

CI/CD流程

CI/CD Pipeline

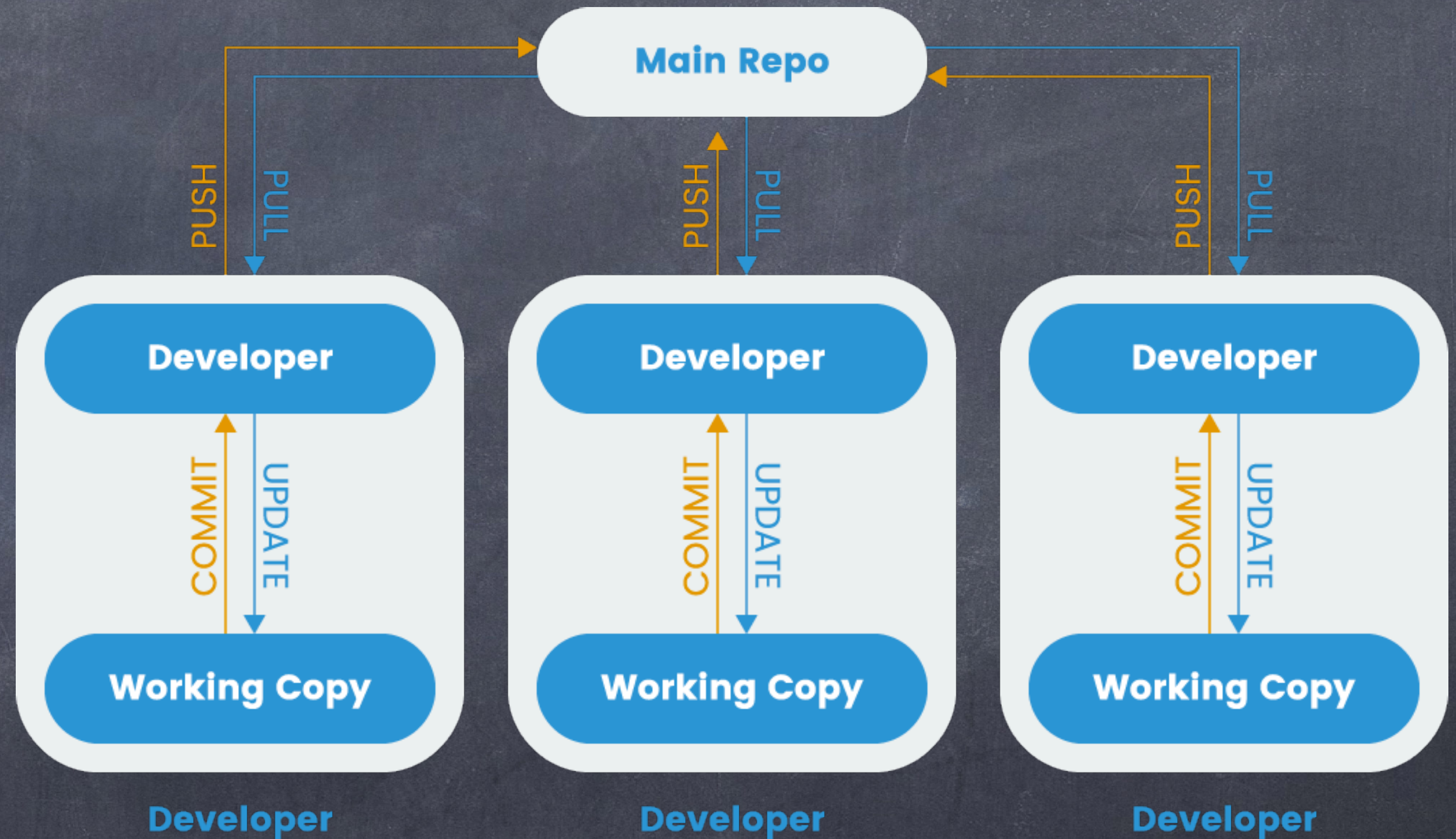


自动构建

- 在软件开发过程中，构建流程会将开发人员生成的代码转换为可执行的可用软件。
 - 对于Go或者C语言等编译语言，此阶段需要通过编译器运行源代码以生成独立的二进制文件。
 - 对于JavaScript或PHP等解释性语言，没有编译的步骤，但是代码依旧需要在特定的时间内冻结、绑定依赖项、打包以便于分发。
 - 这些过程通常称为“构建”或“发布”的工件。
- 虽然开发人员可以手动构建，但这样操作有诸多不利。
 - 首先，从主动开发到创建构建的转变中引入了上下文转换，使得开发人员不得不停止生产效率更高的工作来专注于构建过程。
 - 其次，每个开发人员都在制作自己的工件，这可能导致构建过程不一致。
- 为了解决这些顾虑，许多开发团队配置了自动构建流水线。这些系统监视源代码存储库，并在检测到更改时自动启动预配置的构建过程。这一配置无需牵涉过多的人力在其中并且确保了每个构建过程一致。

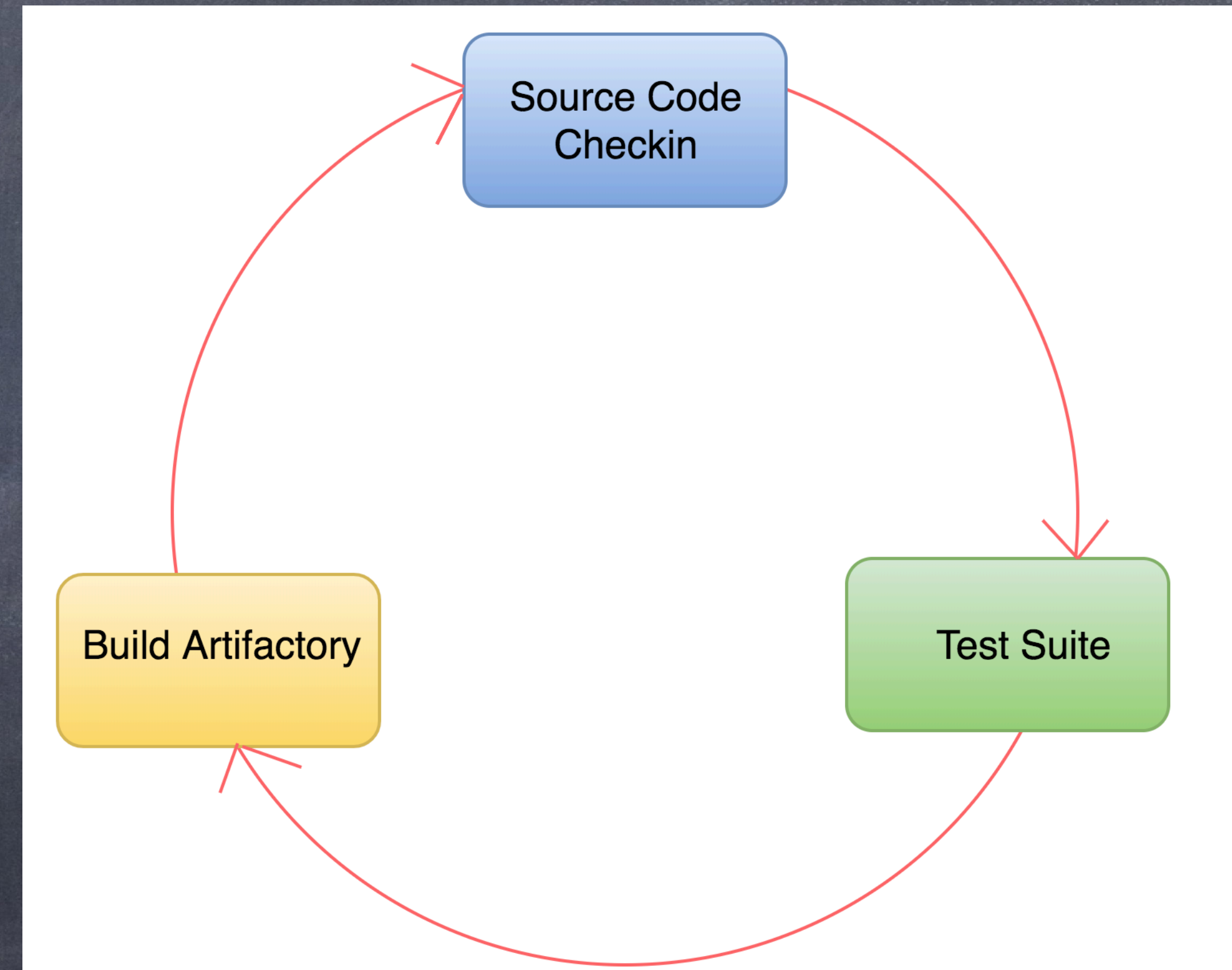
版本控制

- 版本控制系统 (VCS) 用于帮助维护项目历史记录，并行处理离散特征，以及解决存在冲突的更改。VCS允许项目轻松采用更改并在出现问题时回滚。开发人员可以在本地计算机上处理项目，并使用VCS来管理不同的开发分支。
- 记录在VCS中的每个更改都称为提交。每个提交都对代码库的更改进行编目分类，元数据也包含在其中，例如关于查看提交历史记录或合并更新的描述。



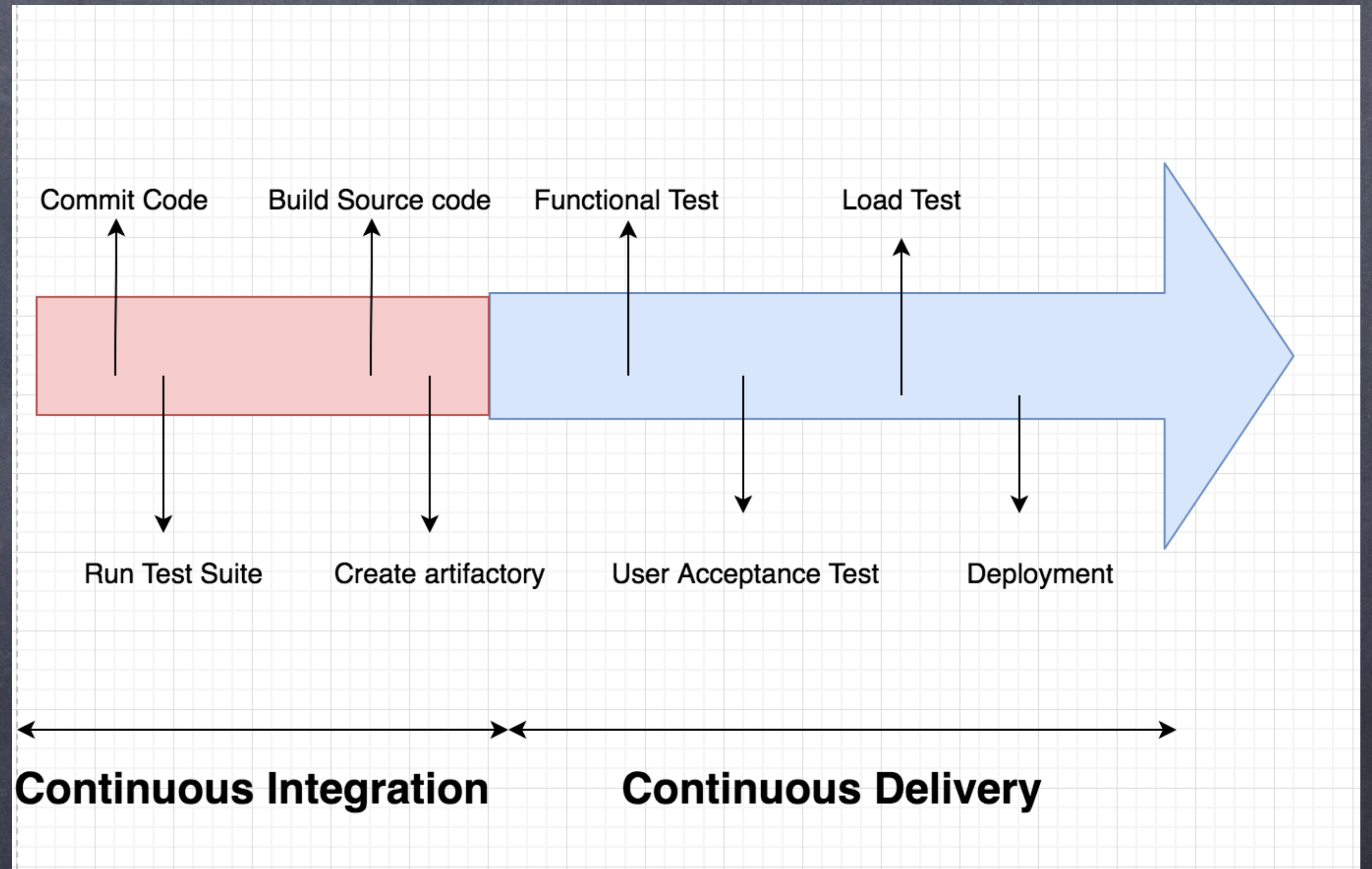
持续集成 (CI)

- 持续集成 (CI) 是一个让开发人员将工作集成到共享分支中的过程，从而增强了协作开发。频繁的集成有助于解决隔离，减少每次提交的大小，以降低合并冲突的可能性。
- 为了鼓励CI实践，一个强大的工具生态已经构建起来。这些系统集成了VCS库，当检测到更改时，可以自动运行构建脚本并且测试套件。集成测试确保不同组件功能可以在一个组内兼容，使得团队可以尽早发现兼容性的bug。因此，持续集成所生产的构建是经过充分测试的，并且是完全可靠的。



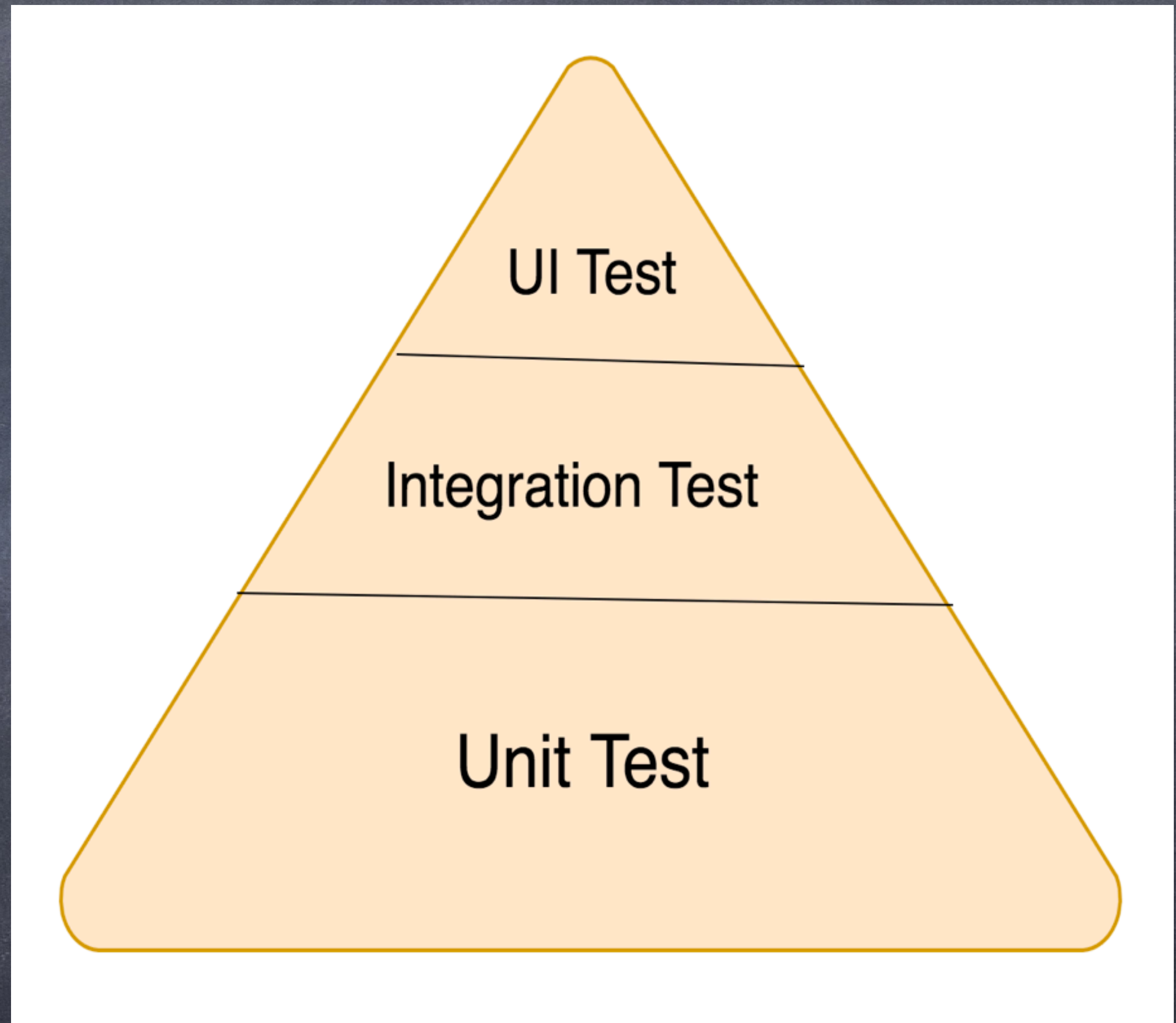
持续交付和持续部署

- 持续交付和持续部署是在构建持续集成的基础之上的两种策略。
- 持续交付 是持续集成的扩展，它将构建从集成测试套件部署到预生产环境。这使得它可以直接在类生产环境中评估每个构建，因此开发人员可以在无需增加任何工作量的情况下，验证bug修复或者测试新特性。一旦部署到staging环境中，就可能需要进行额外的手动和自动测试。
- 持续部署 则更进一步。一旦构建在staging环境中通过了自动测试，持续部署系统将会自动将它部署到生产服务器上。换言之，每个通过测试的构建都是实时的，可供用户及早反馈。这使得团队可以不断发布新特性和修复bug，并以其测试流程提供的保证为后盾。

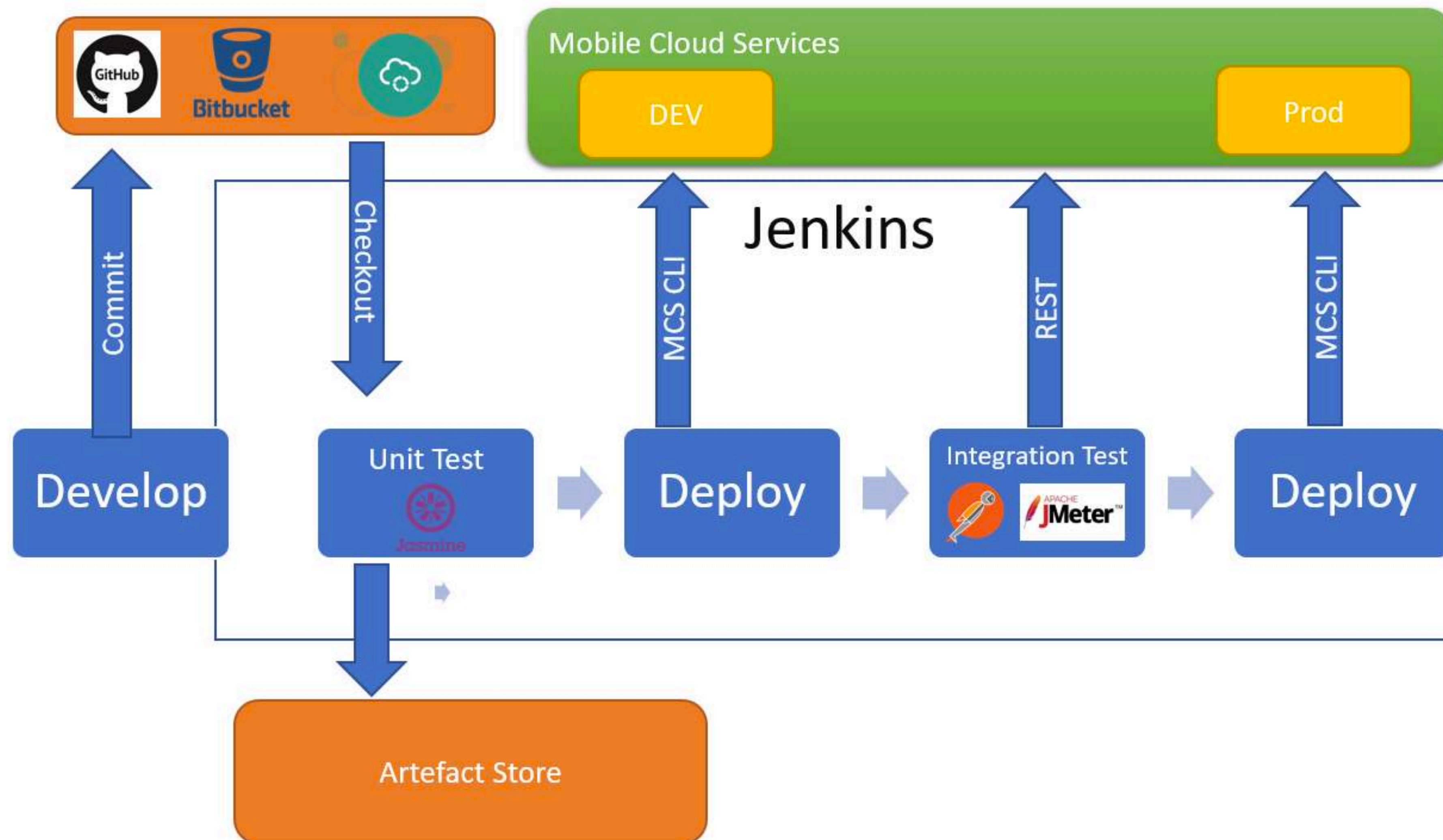


CI/CD的优势

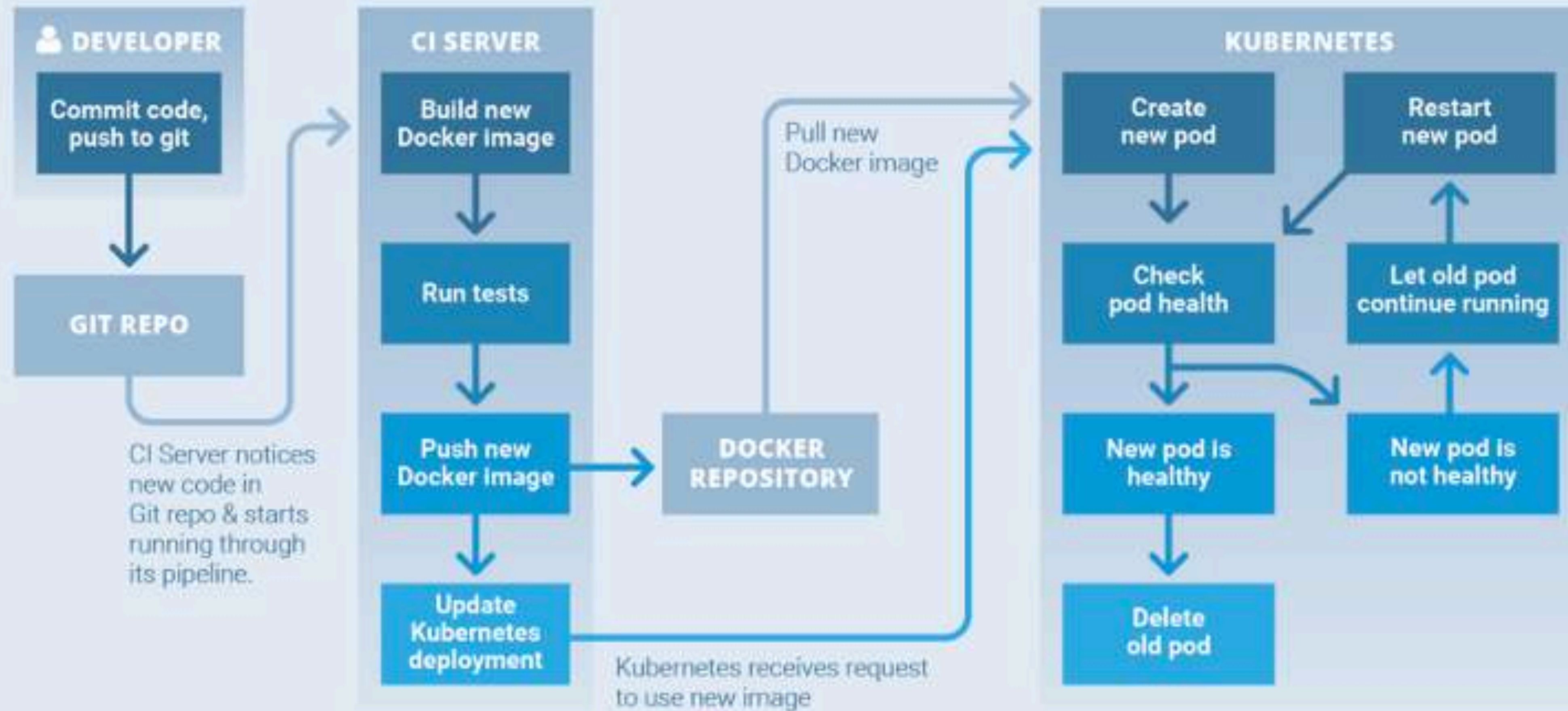
- 快速反馈回路
- 增加可见度
- 简化故障排除
- 软件质量更高
- 集成问题更少
- 有更多的时间专注于开发



CI/CD



CI/CD Pipeline Workflow with Kubernetes



Jenkins 安装过程

